FH SCHMALKALDEN
HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN

Leisefuchs

Analyse eines MIFARE Classic Bezahlsystems

Autoren:
Martin Beier, Beatrice Florat, Jonas Groß, Kai Trott, Volker Zeihs

Kontakt:
dscc@fh-schmalkalden.de

31. Januar 2012

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Danksagung

Wir bedanken uns bei
Herrn Prof. Dr. rer. nat. Dietmar Beyer
für seine Betreuung und Unterstützung,
Herrn Dipl. Informatiker (FH) Lutz Jügelt
für seine technischen Hilfestellungen und
Herrn Prof. Dr. Müller,
der uns in rechtlicher Hinsicht ausgiebig beraten hat.

Außerdem danken wir den gesamten Entwicklern von libnfc, libfreefare und mfcuk.

Unser spezieller Dank gilt auch Anne Scholz, Christian Heinkel und Felix Einsiedel.

1 Einleitung

RFID-Systeme (Radio Frequenz Identifikation) sind allgegenwärtig. Sie kommen in vielen Anwendungen unseres täglichen Lebens, angefangen von elektronischer Produktidentifikation bis hin zu komplexen Zutrittskontrollsystemen, vor.
Gerade in den letzten fünf Jahren wurden in verschiedenen Unternehmen und Behörden etliche Bezahlsysteme, die auf der RFID-Technologie basieren, erfolgreich eingeführt.

Durch zahlreiche wissenschaftliche Veröffentlichungen wurde die Sicherheit dieser Systeme in Frage gestellt. Zu den Arbeiten gehören zum Beispiel „Mifare Classic - Eine Analyse der Implementierung“ [Plö08] und „THE DARK SIDE OF SECURITY BY OBSCURITY“ [Cou09].

Das Ziel dieser Arbeit ist es, im Rahmen eines Forschungsprojektes an der Fachhochschule Schmalkalden, mögliche Schwachstellen eines konkreten Bezahlsystems aufzuzeigen.

Als Zielsystem haben die Autoren ein mittelgroßes Bezahlsystem mit rund einer Million Nutzern ausgewählt, von deren konkreten Nennung aus rechtlichen Gründen abgesehen wird.

2 Grundlagen

In diesem Kapitel werden die grundlegenden Begriffe und Systeme der in dieser Arbeit betrachteten RFID-Technik erläutert. Es wird ein Überblick über die Near Field Communication, die ISO-14443-Norm, die Mifare-Chipkarte und den verwendeten Verschlüsselungsalgorithmus gegeben.

2.1 Einteilung NFC

Near Field Communication (NFC) ist ein Übertragungsstandard zum kontaktlosen Übermitteln von Daten über kurze Strecken. Es gibt zwei Arten der Übertragung, die Verbindungslose und die Verbindungsbehaftete. Die verbindungslosen Systeme können als RFID-Systeme bezeichnet werden.

Man kann bei RFID-Systemen zwischen drei grundsätzlichen Übertragungsmodi unterscheiden. Diese sind

- full-duplex (FDX),
- half-duplex (HDX) und
- sequentielle Systeme (SEQ).

Die Gemeinsamkeit von full- und half-duplex ist, dass die Antwort des Transponders während des eingeschalteten Hochfrequenzfeldes des Lesegerätes übermittelt wird. Bei dem sequentiellen Übertragungsmodus wird das Feld des Lesegerätes periodisch abgeschaltet. In diesem Zeitfenster kann der Transponder die Daten an das Lesegerät übermitteln. Das Problem bei diesem Verfahren besteht jedoch darin, dass durch das Abschalten des Feldes der Transponder nicht mehr mit Energie versorgt werden kann. Dies kann durch den Einbau von Energiespeichern, wie z.B. Kondensatoren, gelöst werden.

Die Transponder haben für verschiedene Anwendungen verschiedene Bauformen: Disks oder Münzen, Glas- oder Plastikgehäuse, Schlüsselanhänger, ID-1 Chipkarte oder Smart Lable. Das gewählte Zielsystem verwendet ID-1-Chipkarten.

In Tabelle 1 kann man die Reichweiten der dabei verwendeten Normen sehen.

Tabelle 1: Reichweiten der Normen [Fin06]

Norm	Kartentyp	Reichweite ca.
ISO/IEC 10536	close coupling	bis 1 cm
ISO/IEC 14443	Proximity coupling	bis 10 cm
ISO/IEC 15693	Vicinity coupling	bis 100 cm

2.2 ISO 14443 Norm

Die ISO/IEC 14443 ist eine internationale Normenreihe für kontaktlose Chipkarten und wurde von der ISO-Working Group 8 (WG8), einer Subgruppe der ISO/IEC SC17, entwickelt. Diese veröffentlicht ihre finalen Normvorschläge unter [wg809] und teilt sie in die folgenden vier Teile ein:

1. physikalische Eigenschaften von PICCs[1] im ID-1-Format
2. Funkinterface zur Energie- und Datenübertragung
3. Rahmenformat für die Kommunikation (Antikollision, Selektion)
4. Datenübertragungsprotokoll

Im Folgenden sollen die einzelnen Teile kurz erläutert werden.

Der erste Teil beschreibt die physikalischen Eigenschaften von PICCs. Er beinhaltet Vorgaben für die Kartengröße, die Resistenz gegen UV- und Röntgenstrahlung, Zug- und Knickfestigkeit, Widerstandsfähigkeit gegen alternierende oder statisch elektrische oder magnetische Felder, sowie die Betriebstemperatur.

Dem PICC wird ein einwandfreies Funktionieren in einem magnetischen Feld von 12 A/m bei 13,56 MHz vorgeschrieben. Außerdem wird die Kompatibilität zu weiteren ISO-Standards, wie z.B. ISO 7811 und ISO 10536, empfohlen.

Der zweite Teil der ISO/IEC 14443 enthält Beschreibungen über das Funkinterface zur Energie- und Datenübertragung. Er spezifiziert die Charakteristiken des Feldes, das die Energie und die bi-direktionale Kommunikation von Daten zwischen dem PCD und den PICCs realisiert (siehe Abbildung 1).

Dieser Teil enthält klar definierte Vorschriften, die den Datenverkehr in den Typen A und B vom und zum PCD regeln, sowie die zur Verwendung zu nutzenden Energiegröße.

[1] Innerhalb der ISO Norm wird das Lesegerät als PCD (proximity coupling device) und die Karte als PICC (proximity integrated circuit card) bezeichnet

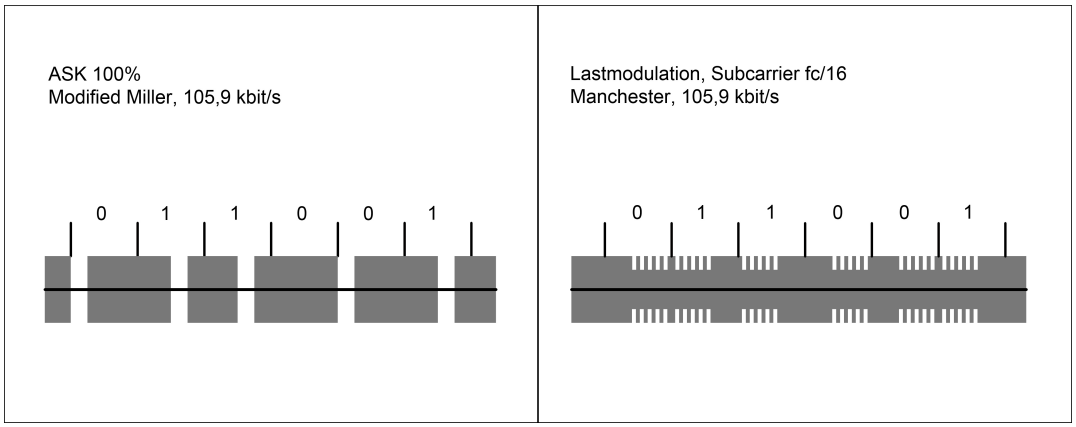

Abbildung 1: Modulation und Kodierung [Plö08]

Typ A arbeitet bei der Datenübertragung von PCD nach PICC mit einer 100% ASK (amplitude shift keying) mit modifizierter Millercodierung. Bei der Übertragung von PICC nach PCD gibt es eine Lastmodulation mit ASK-moduliertem 847,5 kHz Hilfsträger in Manchestercodierung.

Typ B regelt beim Datenverkehr von PCD nach PICC ein Modulationsverfahren von 10% ASK mit NRZ (non return to zero) Codierung. Von PCD nach PICC gibt es eine Lastmodulation mit BPSK (binary phase shift keying) mit moduliertem 847,5 kHz Hilfsträger in NRZ-Codierung.

Der dritte Teil beschreibt das Rahmenformat für die Kommunikation, einen Zustandsautomaten und einen Kommandosatz für die Antikollision und die Selektion. Die Abbildung 2 zeigt den Zustandsautomaten.
Das Antikollisions- und das Selektions-Kommando unterscheiden sich in der Hinsicht, dass bei der Selektion noch zusätzlich die gesamte UID und ein CRC übertragen wird.

Der vierte Teil definiert ein half-duplex blockorientiertes Datenübertragungsprotokoll sowie die Aktivierungs- und Deaktivierungs-Sequenz dieses Protokolls mit den speziellen Bedürfnissen einer kontaktlosen Umwelt.

2.3 Mifare Chipkarte

Die Mifare[2] Chipkarte ist eine kontaktlose Smartcard-Lösung, welche u.a. von NXP und Infineon vertrieben wird. Sie wird unter anderem bei Zutrittskontrollsystemen, bei der Tieridentifikation, bei Bezahlsystemen und bei Ticket-Systemen verwendet.

[2]MIFARE steht für Mikron Far Collection System

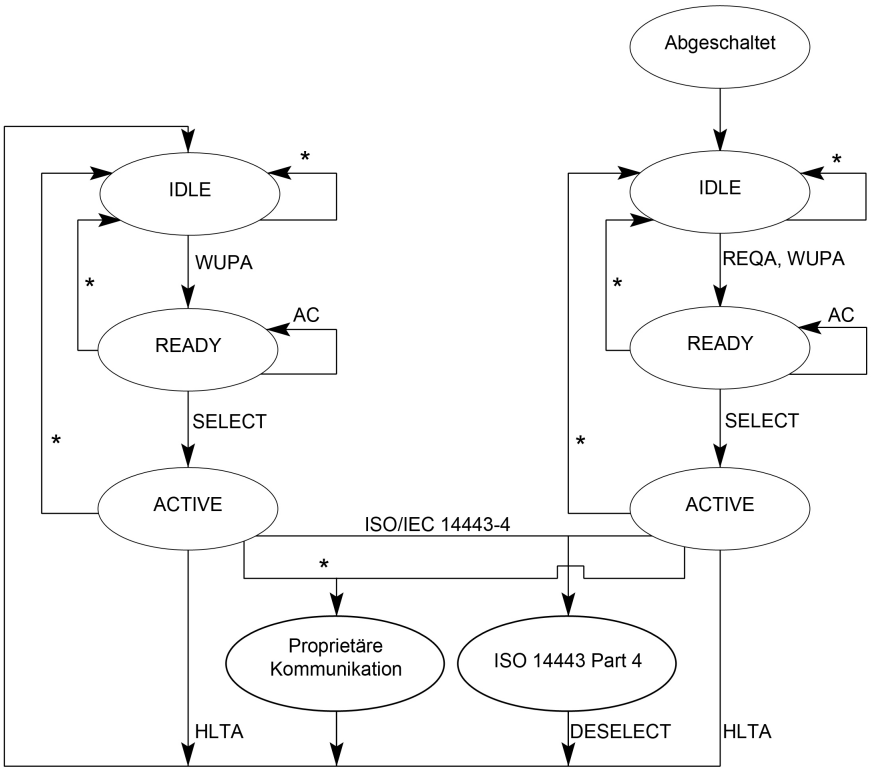

Abbildung 2: Zustandsmaschine für Tags [Plö08]

Ein weiteres Beispiel für die Verwendung der Mifare Technologie ist der neue Personalausweis. Zum Zeitpunkt der Erstellung dieses Dokumentes spricht NXP von über 3,5 Milliarden Mifare-Karten, die im Umlauf sind, und von mehr als 40 Millionen verkauften Lesegeräten (vergleiche [NXP11b]).

Zum Speichern von Daten kommt bei den Mifare PICCs das MAD (Mifare Application Directory) zum Einsatz. Dieses gibt es in drei Versionen, die alle die zu speichernden Daten in sogenannten Applikationen ablegen. Dabei ist die Zweite voll kompatibel zur Ersten. Der einzige signifikante Unterschied ist die Adressierung von einem größeren EEPROM (vergleiche [Sem05] und [NXP09]). Die Mifare-Classic nutzt ausschließlich die Versionen 1 und 2. MAD Version 3 wird in späteren Kartenarten verwendet, wie beispielsweise der Mifare-Desfire. Für diese Arbeit waren nur die Version 1 und 2 von Bedeutung. Der Aufbau des MAD 1 bzw. 2 wird in Abbildung 3 dargestellt.

Das MAD in den Versionen 1 und 2 ist in Sektoren und Blöcke unterteilt. Jeder Sektor besteht aus vier Blöcken, die jeweils 16 Bytes umfassen. Die Blöcke können in Daten- und Trailer-Blöcke eingeteilt werden, wobei die ersten drei Blöcke eines Sektors Datenblöcke sind und der letzte der Trailer-Block ist.

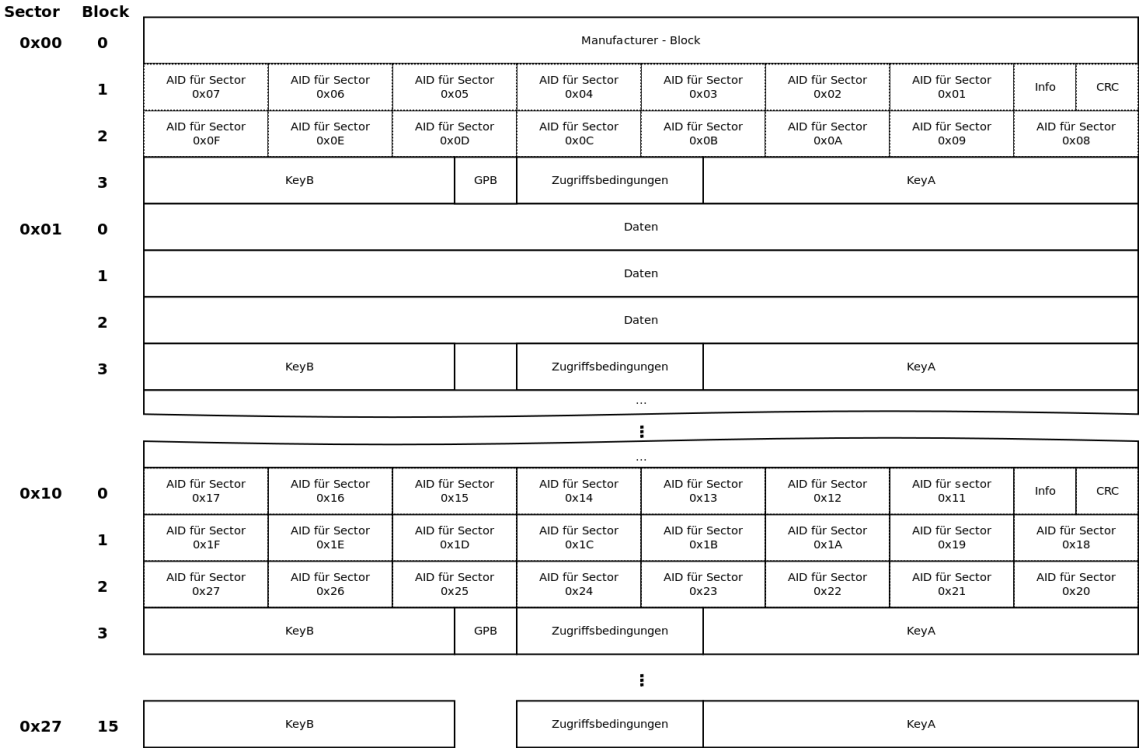

Abbildung 3: Mifare Application Directory Version 2 [Gro11]

Die Datenblöcke werden am Ende mit einer 8 Bit CRC-Checksumme versehen.
Im Trailerblock befinden sich die Read- und Write-Keys, sowie das GPB (General Purpose Byte) und die Zugriffsbedingungen. Das GPB ist lediglich ein Byte, das für eine spätere Verwendung freigehalten wird.

Über den ersten Sektor einer Mifare Smartcard mit MAD 1 oder 2 kann nicht frei verfügt werden. Dieser Sektor enthält die Herstellerdaten im ersten Block, sowie das Inhaltzverzeichnis im zweiten und dritten Block. Das Inhaltsverzeichnis enthält die AIDs (Application Identifier) für die auf der Karte befindlichen Applikationsdaten.

In Version 2 des MAD können nicht nur 1 KB sondern 4 KB verwaltet werden. Dies wird durch ein Verzeichnis ermöglicht, welches in den ersten drei Blöcken des Sektors 16 gespeichert ist.

2.4 Verschlüsselung Crypto1

Crypto1 ist ein proprietärer Verschlüsselungsalgorithmus, der von NXP Semiconductors für ihre Mifare RFID-Produkte entwickelt wurde. Er findet Verwendung in den Mifare-Produkten Classic, Plus und Classic Emulation (in ProX und SmartMX).

Abgesehen von der Mifare Classic Emulation in ProX und SmartMX besitzen die genannten RFID-Produkte einen Hardware-Zufallszahlengenerator (RNG - Random Number Generator). Allerdings basiert der verwendete RNG auf einem linear rückgekoppeltem Schieberegister (LFSR - Linear Feedback Shift Register).
Der Pseudozufallszahlengenerator für die mutual-authentication-Phase ist für die Sicherheit der Verschlüsselung von elementarer Bedeutung. Es ergibt sich eine maximale Periodenlänge von $2^{16} - 1$. Mit der in der ISO-14443 Norm beschriebenen Taktung von 13,56 MHz und einem Frequenzteiler von 128 ergibt sich eine Wiederholung der Zustände des LFSR alle
$\frac{2^{16}-1}{\frac{13,56MHz}{128}} \approx 0,618619s$.

Versuche von Henryk Plötz haben gezeigt, dass es einen Zeitpunkt nach der Aktivierung der Stromversorgung des RFID-Transponders gibt, bei welchem der Zustand des LFSR immer identisch ist. [Plö08] Dies lässt darauf schließen, dass die vom Pseudozufallszahlengenerator erzeugten Zufallszahlen unter umständen nur von der Zeit abhängig sind.

3 Anforderungen

Das Ziel, die Sicherheit des genannten Systems zu kompromittieren, kann als gelungen angesehen werden, wenn

- eine Karte mit manipuliertem
  - Geldbetrag oder
  - Gültigkeitszeitraum

  zum Bezahlen genutzt werden kann und
- die Manipulationen nur zu eigenen Ungunsten durchgeführt werden.

4 Zielsetzung

Es sollte möglich sein, die Anforderungen mit Hilfe von handelsüblichen Geräten und Hilfsmitteln durchzuführen. Im Sinne des Herstellers NXP Semiconductors können bestehende Systeme als gefährdet betrachtet werden, wenn folgende Punkte (Zitat von [NXP11a]) erfüllt sind:

- „ Through overhearing successful communications between the reader of an existing infrastructure and a valid card, the data and/or the keys involved in that transaction could be read
- While overhearing failed communications between the reader of an existing infrastructure and any card, the key used by the reader during that transaction could be retrieved
- These attacks could be carried out in minutes or less and with means involving a laptop and equipment which can be built with limited material cost (100 Euros)
- Card only attacks are possible in lab environments and at considerable precalculation time. This is expected to further evolve into an attack that does not need lab conditions and may require less precalculation time.
- In one particular "card only" attack, all keys and data can be retrieved within seconds using a laptop and some low value equipments. In this attack, the attacker needs to have a certain knowledge about the card.“

Das Projektteam hat, unter Zuhilfenahme dieser Richtlinien von NXP Semiconductors, folgende Anforderungen für sich definiert:

- die Kosten für die benötigten Zusatzgeräte sollten unter 100 € liegen,
- Veränderungen sind nur auf der eigenen Karte und nicht auf dem restlichen System durchzuführen,
- die Manipulation wird nur zu den eigenen Ungunsten getätigt und
- es werden keine fremden Daten Ausgespäht.

Nach ausgiebiger rechtlicher Beratung ist das Projektteam zu dem Schluss gekommen, dass es auf legalem Wege nicht möglich ist, beliebige Manipulation auf der Karte durchzuführen. Aus diesem Grunde wurden auch keine fremden Daten ausgespäht.

5 Verwendetes Equipment

Für die Durchführung der Tests wurde folgende Ausrüstung genutzt:

Die benötigte Hardware bestand aus: dem RFID-Lesegerät Touchatag, einer gültigen Karte des Bezahlsystems und einem tragbaren Computer.

Die verwendetete Software bestand sowohl aus Open-Source-Projekten, sowie selbst erstellten Programmen bzw. Skripten, die im Folgenden aufgezählt werden.

Es wurden zwei Beispielprogramme der libnfc verwendet. Zum Einen gehört das nfc-mfclassic dazu, das zum Erzeugen eines Abbildes einer Mifare-Classic-Karte verwendet wird. Zum Anderen wird zum Bestimmen des Kartentyps das Programm nfc-anticol benutzt.

Die in der libfreefare enthaltenen Dienstprogramme mfoc und nfc-mfformat sind nötig für das Herausfinden der Schlüssel und das Löschen einer Karte. Daneben wird mfcuk für das Schließen auf einen verwendeten Schlüssel benötigt.

Die selbst-entwickelte Software bestand aus einem Parser zur Anpassung des Trailerblocks und einem Skript, welches die Checksummen für jeden Block berechnet.

6 Analyse des Systems

Das betrachtete System verwendet Karten nach dem Mifare-Classic-Standard von den Herstellern NXP Semiconductors und Infineon. Da es für diesen Kartentyp im Vorfeld bekannte Angriffsszenarien gab, wurden diese näher betrachtet. Dabei stießen die Autoren auf zwei Angriffsmöglichkeiten, die sich als sinnvoll erweisen könnten. Zum Einen ist das mfoc, das in der libfreefare enthalten ist. Zum Anderen ist das darkside, das in mfcuk vorhanden ist.

Diese Tools versuchen die Zugriffsschlüssel mit Hilfe verschiedener kryptographischer und statistischer Methoden und Verfahren wiederherzustellen.

Das betrachtete Bezahlsystem ist auf mehrere Standorte in Thüringen verteilt. Da die geplanten Schritte nur mit einer MIFARE Classic-Karte durchzuführen waren und dies die weltweit meist verwendete Chip-Karten-Technologie ist, wurde ein Standort gewählt, der diese benutzt.

Zunächst wurde mfoc an einer Karte des betrachteten Bezahlsystems angewendet. Es stellte sich heraus, dass die Karte keine Standardschlüssel verwendet. Es war nicht möglich mittels mfoc auf die Schlüssel zu schließen. Daher wurde versucht mit Hilfe von mfcuk die Schlüssel herauszufinden.

Da bisher das mfcuk noch nicht auf eine aktuelle Version von libnfc angepasst wurde, musste dies zunächst implementiert werden. Nach Abschluss dieses Vorgangs war es möglich mit mfcuk einen der 32 Zugriffsschlüssel zu ermitteln. Mit der verwendeten Hardware dauerte dies ca. eine Stunde. Nun konnte mfoc eingesetzt werden, um auf die anderen Schlüssel zu schließen. Nach weiteren acht Stunden waren alle 32 Zugriffsschlüssel bekannt.

Um die Originalkarte zu kopieren und das Geld auf dieser Karte nicht zu verdoppeln, wurde diese zunächst geleert. Das heißt, von dem noch vorhandenem Geld wurde etwas gekauft, das demselben Wert entspricht. Danach wurde von der Originalkarte mit einem aktuellen Guthaben von 0 € ein Abbild erstellt.

Um dieses Abbild auf die leere Karte zu speichern, mussten erst die Access-Bits auf diesem Abbild geändert werden. Die Access-Bits waren so gesetzt, dass nur ein einmaliges Beschreiben eines Sektors möglich war. Zur Beschleunigung wurde ein Programm[3] geschrieben, das dies automatisiert.

Als nächster Schritt wurde versucht auf die kopierte Karte Bargeld mittels dafür vorgesehenem Auflade-Automaten aufzuladen. Der Vorgang löste eine Fehlermeldung aus. Vermutlich, weil beim Aufladevorgang die nicht änderbare Unique Identification Number (UID) überprüft wird. Vor Ort war eine Prüfung nicht möglich. Dagegen führte die Verwendung der Originalkarte am gleichen Automaten zum Aufladevorgang.

Um die Zielsetzung dennoch zu erfüllen, wurden alle weiteren Manipulationen auf der Originalkarte des Bezahlssystems vorgenommen. Da, aus rechtlichen Gründen, das Projektteam nur zu eigenen Ungunsten manipulieren wollte, wurden 5 € dekrementiert. Nun wurde mit der manipulierten Karte etwas gekauft, um zu testen, ob die nicht vom System vorgenommenen Änderungen an der Karte bemerkt werden. Der Test war in soweit erfolgreich, da die Änderungen nicht bemerkt wurden. Im nächsten Schritt wurden die zuvor abgezogenen 5 € wieder inkrementiert, um auch das Erhöhen eines Geldbetrages zu untersuchen. Dies gelang ebenfalls ohne Komplikationen.

[3] parser.c

Das Inkrementieren und Dekrementieren wurde wie von einem gültigen Gerät des Bezahlsystems mit der Veränderung aller dazugehörigen Bereiche vorgenommen. Zu diesen Bereichen zählen u.a. der Geldbetrag mit all seinen Speicherorten, der Auflade- bzw. Bezahlvorgang und die dazugehörigen Prüfsummen.

Nachfolgend sind die Schritte in einer übersichtlicheren Kurzform beschrieben:

1. Zugangsschlüssel ermitteln
2. Original-RFID-Karte leeren (Geldbetrag nullen)
3. Abbild der Original-Karte erstellen
4. Originalkarte auf eine leere Karte kopieren
5. kopierte Karte überprüfen
6. 10 € am Auflader auf kopierte Karte aufladen
7. Aufladung auf kopierte Karte fehlgeschlagen
8. 10 € auf Originalkarte aufgeladen
9. Abbild der Originalkarte erstellt
10. 5 € auf dem Abbild dekrementieren
11. dieses Abbild auf Originalkarte aufspielen
12. Originalkarte überprüfen
13. mit Originalkarte etwas kaufen
14. Abbild der Originalkarte erstellen
15. vorher abgezogenes Geld (5 €) auf dem Abbild inkrementieren
16. dieses Abbild auf Originalkarte aufspielen
17. Originalkarte überprüfen
18. mit Originalkarte etwas kaufen
19. Abbild der Originalkarte erstellen
20. ERFOLG

7 Ergebnis

Der Test war erfolgreich. Dem Projektteam ist es gelungen, mit handelsüblicher Hardware und frei verfügbarer Software,inklusive eigener Anpassungen und Programme auf einer Karte des Bezahlsystems den Geldbetrag und den Gültigkeitszeitraum zu manipulieren. Die angestrebte Manipulation zu eigenen Ungunsten war möglich.

Mit Hilfe der bekannten Keys konnte die Karte ausgelesen und ein Abbild erstellt werden. Durch gezielten Einsatz der Karte und immer wieder erstellter Abbilder konnten mittels Differenzanalyse die einzelnen Bereiche eingegrenzt werden. An unterschiedlichen Kassen und Aufladern wurden verschiedenste Transaktionen durchgeführt. Dabei waren vier Karten im Einsatz. Pro Karte wurden 20 - 30 Transaktionen getätigt. Dabei ergaben sich in fünf verschiedenen Sektoren Veränderungen.

Diese sind

- das aktuelle Guthaben (0x40 - 0x5F),
- der letzte Aufladevorgang (0x80 - 0x8F) und der letzte Bezahlvorgang (0x90 - 0x9F),
- das Ende des Gültigkeitszeitraums (0xC0 - 0xCF) und der Anfang des Gültigkeitszeitraums (0xE0 - 0xEF),
- die Kartennummer in umgekehrter Reihenfolge (0x100 - 0x11F) und
- die Kartennummer (0x140 - 0x14F).

7.1 Kodierungsverfahren

Durch umfangreiches empirisches Analysieren der Binärdateien (Abbilder der Karten) mittels eines Hexeditors konnten die oben genannten Werte und die verschiedenen Kodierungsarten ermittelt werden. Dabei stellte sich heraus, dass die unterschiedlichsten Kodierungsverfahren angewendet wurden. Weiterhin wurden die Daten redundant gespeichert.

Die folgenden Kodierungsverfahren wurden erkannt:

- Little Endian (siehe Tabelle 2),
- Inversion (siehe Tabelle 2),
- Big Endian (siehe Tabelle 3),
- ASCII (siehe Tabellen 6 und 7) und
- andere Mischkombinationen (siehe Tabellen 4 und 5).

7.2 Analyse des Abbilds

Tabelle 2: Aktuelles Guthaben

Adr.	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0040:	11	03	00	00	EE	FC	FF	FF	11	03	00	00	00	FF	00	FF
0050:	11	03	00	00	EE	FC	FF	FF	11	03	00	00	00	FF	00	FF

In Tabelle 2 in den Bytes $40_{16} - 41_{16}$ und $48_{16} - 49_{16}$ befindet sich der Geldbetrag in Eurocent in Little Endian-Notation gespeichert. In Byte 44_{16} und 45_{16} ist dieser zusätzlich invertiert abgelegt. Die Bytes $40_{16} - 4F_{16}$ werden redundant in den Bytes $50_{16} - 5F_{16}$ abgelegt. Im Beispiel kann man sehen, wie sich die 785 Eurocent des aktuellen Guthabens berechnen lassen.

$1103_{16} \xrightarrow{LittleEndian} 0311_{16}\ entspricht\ 785_{10}\ Eurocent$

$EEFC_{16} \xrightarrow{invertiert} 1103_{16} \xrightarrow{LittleEndian} 0311_{16}\ entspricht\ 785_{10}\ Eurocent$

Tabelle 3: Auflade- und Bezahlvorgang

Adr.	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0080:	20	29	16	03	07	DB	09	28	00	00	01	F4	00	00	00	EB
0090:	26	01	16	03	07	DB	09	10	00	00	00	00	00	00	24	2C

In Tabelle 3 ist der Auflade- und Bezahlvorgang zu sehen. Die Bytes $82_{16} - 87_{16}$ werden vom Aufladegerät gesetzt. Dabei ist Byte 82_{16} der Tag, Byte 83_{16} der Monat, Byte 84_{16} und 85_{16} das Jahr, Byte 86_{16} ist die Stunde und Byte 87_{16} ist die Minute. Von Byte $8A_{16} - 8B_{16}$ ist der aufgeladene Betrag in Eurocent gespeichert. In Byte $8F_{16}$ befindet sich die Checksumme. In Kapitel 7.3 wird die Berechnung der Prüfsummen beschrieben.

$16_{16} \rightarrow 24_{10}\ Tag$
$03_{16} \rightarrow 03_{10}\ Monat$
$07DB_{16} \rightarrow 2011_{10}\ Jahr$
$28_{16} \rightarrow 40_{10}\ Minute$
$01F4_{16} \rightarrow 500_{10}\ aufgeladener\ Betrag\ in\ Eurocent$

Die Bytes $92_{16} - 97_{16}$ werden von der Kasse gesetzt, Byte 92_{16} ist der Tag, Byte 93_{16} ist der Monat, Byte 94_{16} und 95_{16} ist das Jahr, Byte 96_{16} ist die Stunde und Byte 97_{16} ist die Minute. Von Byte $9A_{16} - 9B_{16}$ ist der abgebuchte Betrag in Eurocent gespeichert, jedoch wird dieser nicht von jeder Kasse abgelegt. In Byte $9F_{16}$ befindet sich die Checksumme.

$16_{16} \rightarrow 24_{10}\ Tag$
$03_{16} \rightarrow 03_{10}\ Monat$
$07DB_{16} \rightarrow 2011_{10}\ Jahr$
$09_{16} \rightarrow 09_{10}\ Stunde$
$10_{16} \rightarrow 16_{10}\ Minute$

Tabelle 4: Ende des Gültigkeitszeitraumes

Adr.	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00C0:	31	03	20	11	32	00	01	00	C8	00	00	00	00	30	00	37

In Tabelle 4 ist das Ende des Gültigkeitszeitraumes zu sehen. Von Byte $C0_{16}$ bis $C3_{16}$ ist das Datum in der Form abgelegt: Tag, Monat, Jahr. Dieses ist in hexadezimalem Format wie Dezimal zu lesen. Im Byte CF_{16} befindet sich die Checksumme.

$31032011_{16} \rightarrow 31032011_{10}$ *entspricht dem* 31.03.2011

Tabelle 5: Anfang des Gültigkeitszeitraumes

Adr.	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00E0:	11	04	20	05	34	D4	85	6B	B9	00	00	00	00	00	00	78

In der Tabelle 5 kann man den Anfang des Gültigkeitszeitraumes sehen. In den Bytes $E0_{16} - E3_{16}$ ist das Datum in folgendem Format abgelegt: Tag, Monat, Jahr. Dieses ist in hexadezimalem Format wie dezimal zu lesen. In Byte EF_{16} befindet sich die Checksumme.

$11042005_{16} \rightarrow 11042005_{10}$ *entspricht dem* 11.04.2005

Tabelle 6: Kartennummer in umgekehrter Reihenfolge

Adr.	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0100:	30	39	39	39	39	39	39	39	31	30	34	27	07	08	99	73
0110:	30	39	39	39	39	39	39	39	31	30	34	27	07	08	99	73

In Byte $101_{16} - 10A_{16}$ wurde die umgedrehte Kartennummer gespeichert (siehe Tabelle 6). Diese ist in ASCII-Notation abgelegt, wie im nachfolgenden Beispiel gezeigt wird. In Byte $10F_{16}$ ist die Checksumme abgelegt. Dieser Block ist redundant gespeichert in den Bytes $110_{16} - 11F_{16}$.

$39_{16} \xrightarrow{ASCII} 9$
$39_{16} \xrightarrow{ASCII} 9$
$39_{16} \xrightarrow{ASCII} 9$
$39_{16} \xrightarrow{ASCII} 9$
$39_{16} \xrightarrow{ASCII} 9$
$39_{16} \xrightarrow{ASCII} 9$
$39_{16} \xrightarrow{ASCII} 9$
$31_{16} \xrightarrow{ASCII} 1$
$30_{16} \xrightarrow{ASCII} 0$
$34_{16} \xrightarrow{ASCII} 4$
gedreht : 4019999999

Tabelle 7: Kartennummer

Adr.	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0140:	33	34	30	31	39	39	39	39	39	39	39	34	30	30	00	F5

In den Bytes $141_{16} - 14A_{16}$ ist die Kartennummer abgelegt (siehe Tabelle 7). Diese ist in ASCII-Notation gespeichert. In Byte $14F_{16}$ befindet sich die Checksumme.

$34_{16} \xrightarrow{ASCII} 4$
$30_{16} \xrightarrow{ASCII} 0$
$31_{16} \xrightarrow{ASCII} 1$
$39_{16} \xrightarrow{ASCII} 9$
$39_{16} \xrightarrow{ASCII} 9$
$39_{16} \xrightarrow{ASCII} 9$
$39_{16} \xrightarrow{ASCII} 9$
$39_{16} \xrightarrow{ASCII} 9$
$39_{16} \xrightarrow{ASCII} 9$
$39_{16} \xrightarrow{ASCII} 9$
entspricht : 4019999999

7.3 Prüfsumme

Die Prüfsumme ist ein Byte, das am Ende eines Blockes steht. Diese ist u.a. in den Sektoren des Auflade- und Bezahlvorgangs, des Gültigkeitszeitraumes und der Kartennummer zu finden. Die Prüfsumme berechnet sich, indem die ersten 15 Bytes eines Blocks mit XOR verknüpft werden und dieses Ergebnis anschließend invertiert wird.

Die Checksumme wird in allen Beispielen wie folgt berechnet. Gezeigt wird dies am Beispiel von Block 21, beginnend bei Byte 140_{16} (Tabelle 7).

$33_{16} \oplus 34_{16} \oplus 30_{16} \oplus 31_{16} \oplus 39_{16} \oplus 39_{16} \oplus 39_{16} \oplus 30_{16} \oplus 34_{16} \oplus 33_{16} \oplus 36_{16} \oplus 34_{16} \oplus 30_{16} \oplus 30_{16} \oplus 00_{16}$
$\rightarrow 0A \xrightarrow{invertiert} F5$

8 Evaluation

Die Auswertung der Analysen und Tests des Bezahlsystems ergibt, dass man dieses ohne größeren Aufwand und mit geringem Einsatz manipulieren kann. Es ist besorgniserregend, dass die Tests, die von den Autoren durchgeführt wurden, schon seit längerem den Herstellern und deren Kunden bekannt sind. Desweiteren wurde festgestellt, dass die zuvor beschriebenen Angriffe nicht systemspezifisch, sondern nur abhängig von dem verwendeten RFID-Chip sind. Im hiesigen Fall, ist es ein MIFARE-Classic Chip. Dadurch ergeben sich Angriffe auf die verschiedensten RFID-Systeme, beispielsweise Nahverkehrssysteme, Zutrittskontrollsysteme und Identifikationssysteme.

9 Zukunftsbetrachtung

Bei der Durchführung der Angriffe ist, bei dem Versuch Geld auf eine kopierte Karte mit einem Auflader aufzuladen, ein Fehler aufgetreten. Es wird vermutet, dass beim Aufladevorgang die UID der Karte überprüft wird. Es ist nicht möglich diese direkt auf der RFID-Chipkarte zu ändern.

Um diesen Sicherheitsmechanismus zu umgehen, könnte man einen RFID-Chip emulieren. Dieses ist derzeit nur sehr eingeschränkt mit dem benutzten Lesegerät, dem Touchatag möglich. Daher wurde das Emulieren nicht näher betrachtet.

Zusätzlich ist bei dem untersuchten System auch folgender Test denkbar: die Manipulation der Zutrittskontrolle für Kfz-Stellplätze und für Gebäude und Räume.

Es ist lobenswert, dass einige Anbieter solcher Systeme bei der Entwicklung neuer Systeme dies berücksichtigen und Chips mit einem offenen Verschlüsselungsverfahren einsetzen. In der Vergangenheit hat sich gezeigt, dass offene Verschlüsselungsstandards den proprietären meist überlegen waren. Karten mit solchen offengelegten Standards sind zum Beispiel Mifare Plus, Mifare Desfire und Mifare Desfire EV1. Diese sind jedoch preisintensiver und schrecken daher noch einige Unternehemen vom Einsatz dieser ab. Es empfiehlt sich dennoch die neueren und sichereren Standards einzusetzen.

Literatur

[Cou09] Courtois, Nicolas T.: *THE DARK SIDE OF SECURITY BY OBSCURITY*. Technischer Bericht, University College London, 2009.

[Fin06] Finkenzeller, Klaus: *RFID-Handbuch*. Hanser Verlag, 4. Auflage, 2006.

[Gro11] Gross, Jonas: *Mifare DESfire Eine Analyse der Implementierung*. Diplomarbeit, FH Schmalkalden, 2011.

[NXP09] NXP: *MIFARE Application Directory (MAD)*. Technischer Bericht, NXP B.V., 2009. `www.nxp.com/acrobat_download2/other/identification/AN10787_6.pdf` 30. September 2010.

[NXP11a] NXP: *MIFARE classic vulnerabilities*. Technischer Bericht, NXP Semiconductors Austria GmbH Styria, 2011. `http://mifare.net/technology/security/mifare-classic/` 20. Juli 2011.

[NXP11b] NXP: *Overview*. Technischer Bericht, NXP Semiconductors Austria GmbH Styria, 2011. `http://mifare.net/overview/` 26. Juni 2011.

[Plö08] Plötz, H.: *Mifare Classic–Eine Analyse der Implementierung*. Master's thesis, Humboldt-Universität zu Berlin, 2008.

[Sem05] Semiconductors, Philips: *MIFARE DESFire MF3 IC D40 Short Form Specification*. Technischer Bericht, Koninklijke Philips Electronics N.V., 2005. `www.nxp.com/documents/short_data_sheet/075532.pdf` 30. September 2010.

[wg809] wg8: *Standing Document 1*. Technischer Bericht, Working Group 8, 2009. `http://wg8.de/sd1.html` 30. Juni 2011.