Humboldt-Universität zu Berlin
Mathematisch-Naturwissenschaftliche Fakultät
Institut für Informatik

eID-Funktion „in the middle“

Bachelorarbeit

zur Erlangung des akademischen Grades
Bachelor of Science (B. Sc.)

eingereicht von: Malte Schmidt
geboren am: 20.9.1989
geboren in: Berlin

Gutachter/innen: Prof. Dr. Jens-Peter Redlich
Prof. Dr. Ernst-Günter Giessmann

eingereicht am: ..... verteidigt am: .....

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

nPA neuer Personalausweis

eID elektronischer Identitätsnachweis

VfB Vergabestelle für Berechtigungszertifikate

BSI Bundesamt für Sicherheit in der Informationstechnik

PACE Password Authenticated Connection Establishment

EAC Extended Access Control

TA Terminal Authentication

CA Chip Authentication

APDU Application Protocol Data Units

rAPDU response APDU

cAPDU command APDU

ATR Answer to Reset

BT Bluetooth

LE Low Energie

SAP SIM Access Profile

1. Einleitung

Der neue Personalausweis (nPA) ist seit dem 1. November 2010 als amtlicher Lichtbildausweis der Bundesrepublik Deutschland im Umlauf. Einem aktuellen Bericht des Bundesministeriums des Inneren zufolge besitzen rund 40 Millionen Bürger bereits den neuen Ausweis [1]. Die Online-Ausweisfunktion „elektronischer Identitätsnachweis (eID)“ ist allerdings nur etwa bei jedem Dritten aktiviert. Das Bundesinnenministerium kann keine genauen Aussagen zur Nutzung der Online-Ausweisfunktion machen. Eine Studie der Gesellschaft für Konsumforschung (GfK SE) im Auftrag der Welt am Sonntag [2] zeigt, dass lediglich 5 Prozent der Bürger im letzten Jahr die eID-Funktion genutzt haben.
Es liegt die Vermutung nahe, dass für viele Bürger die eID-Funktion nicht reizvoll genug ist um die Anschaffungskosten für ein Kartenlesegerät zu rechtfertigen, welches für die Nutzung der eID-Funktion notwendig ist.
Nicht nur die Anzahl der Nutzer der Online-Ausweisfunktion ist gering. Auch die Anzahl an Diensteanbietern ist noch sehr beschränkt. So beinhaltet die von der Vergabestelle für Berechtigungszertifikate (VfB) veröffentlichte Liste der erteilten Berechtigungszertifikate für Diensteanbieter lediglich 249 Einträge [3]. Es ist denkbar, dass es hier eine Wechselwirkung gibt: Gibt es nur wenige Dienstanbieter und Dienste, ist der Anreiz von der eID-Funktion für den Bürger gering. Nutzen wiederum nur wenige Bürger den elektronischen Identitätsnachweis, erscheint es potentiellen Dienstanbietern möglicherweise auch nicht lohnenswert, einen Dienst zur Verfügung zu stellen. Auch wird nicht jeder Dienstanbieter die Anforderungen der VfB für das Berechtigungszertifikat erfüllen können. Von dort heißt es: „Diensteanbieter, können sich vorbereiten, indem sie die eigenen Geschäftsprozesse analysieren und dabei die Nutzung der Daten innerhalb des definierten Geschäftsprozesses darlegen. Für die VfB ist dies das entscheidende Kriterium bei der Prüfung des Antrags.“ [12]

Theoretisch ist es möglich auf das Berechtigungszertifikat und einen eigenen eID-Server zu verzichten und stattdessen dem Nutzer bei einer Selbstauskunft logisch über die Schulter zu schauen.
Dies würde den Aufwand für potentielle Dienstanbieter verringern, aber welche Konsequenzen ergeben sich dabei für die Nutzer?
Ziel dieser Arbeit ist es, dieser Möglichkeit nachzugehen. Es wird eine Android-App entwickelt, die die Nutzung der eID-Funktion des neuer Personalausweis (nPA) „in the middle“ ermöglicht, d.h. eine Selbstauskunft aufzurufen und die vertrauenswürdig ausgelesenen Ergebnisse zu nutzen.

Die für das Thema notwendigen Grundlagen werden in Kapitel 2 beschrieben. Die dazu eingesetzte Hard- und Software, sowie die Lösungsstrategie für die Implementierung werden in Kapitel 3 erläutert. Anschließend werden die Ergebnisse in Kapitel 4 vorgestellt und in Kapitel 5 die erhaltenen Ergebnisse und ihre Implikationen diskutiert.

2. Grundlagen

2.1. Der neue Personalausweis

Der neue Personalausweis (nPA) besitzt neben dem neuen Format auch einen Chip mit drei elektronischen Funktionen: die Online-Ausweisfunktion (eID), die Unterschriftsfunktion (eSign) und die Biometriefunktion (ePass). Der neue Personalausweis ist wie der bisherige auch ein Sichtausweis. Die folgenden Daten des Inhabers sind lesbar aufgedruckt:

- Name
- Geburtsname[1]
- Vornamen
- Doktorgrad
- Geburtstag
- Geburtsort
- Staatsangehörigkeit
- Lichtbild
- Unterschrift
- Augenfarbe
- Größe
- Anschrift[2]
- Ggf. Ordens- oder Künstlername
- Seriennummer
- Card Access Number (CAN)

Von diesen Daten werden die eigenhändige Unterschrift, die Größe sowie die Augenfarbe nicht auf dem Chip gespeichert.
Von den gespeicherten Daten lassen sich nach Zustimmung des Inhabers mit eID-PIN-Eingabe die meisten Daten übermitteln und für die Online-Ausweisfunktion nutzen. Lediglich das Lichtbild, die Seriennummer des Ausweises und, falls gespeichert, die Fingerabdrücke sind, wie in Abbildung 1 zu sehen, ausschließlich für die hoheitliche Identitätskontrolle vorgesehen und lassen sich somit nicht für die Online-Ausweisfunktion nutzen.

Damit sich ein Karteninhaber online ausweisen kann, wird beim Diensteanbieter ein eID-Server benötigt. Ein Anbieter eines solchen Dienstes benötigt ein Berechtigungszertifikat von der Vergabestelle für Berechtigungszertifikate. Für die Erteilung eines solchen muss der Anbieter einige Bedingungen erfüllen.
Dazu gehört unter anderem ein Nachweis der Erforderlichkeit der Datenfelder nach §18 Abs. 3 PAuswG., sowie Erklärungen zu Datenschutz und Datensicherheit.
Nach dem Erhalt eines Berechtigungszertifikats kann der Diensteanbieter entweder einen eigenen eID-Server betreiben oder einen eID-Service-Provider nutzen.

[1] seit November 2013
[2] bei Anschrift im Ausland die Angabe „keine Hauptwohnung in Deutschland“

Bei welcher Aktion dürfen welche Daten übertragen werden?

Daten	Hoheitliche Identitätskontrolle	Online-Ausweisfunktion (freiwillig)
Familienname und Vornamen	✓	✓
Geburtsdatum und -ort	✓	✓
Anschrift und Postleitzahl	✓	✓
wenn angegeben: Ordens- bzw. Künstlername	✓	✓
wenn angegeben: Doktorgrad	✓	✓
Biometrische Daten		
digitales Lichtbild	✓	✗
wenn gewünscht: digitale Fingerabdrücke	✓	✗
Weitere Angaben		
Seriennummer des Ausweises	✓	✗

Abbildung 1: Bei welcher Aktion dürfen welche Daten übertragen werden [4]?

2.1.1. Infrastruktur bei Bürgerinnen und Bürgern

Um sich mit dem nPA bei einem Dienst im Internet zu authentifizieren, werden folgende Komponenten beim Nutzer benötigt:

- ein neuer Personalausweis mit aktivierter eID-Funktion
- ein Kartenlesegerät nach TR-03119 (siehe Kapitel 2.4)
- die sechstellige eID-PIN des Nutzers

Außerdem wird eine Software benötigt, mit der eine sichere Verbindung zwischen Kartenleser, nPA und Diensteanbieter hergestellt werden kann.
Seit dem 1. November 2014 gibt es dafür die quelloffene *AusweisApp2* [5]. Während die *AusweisApp* mit Version 1.2 noch Linux-Distributionen unterstützte, wurde bei der *AusweisApp2* darauf verzichtet. Für Linux-Nutzer gibt es allerdings Open-Source-Clients, wie beispielsweise *PersoApp* [6] oder *Open eCard App* [7].

2.2. Die Selbstauskunft

Die Selbstauskunft [8] bietet die Möglichkeit, den neuen Personalausweis auszuprobieren oder zu sehen, welche Daten auf dem Chip gespeichert sind und über die eID-Funktion ausgelesen werden können. Abbildung 2 zeigt den Ablauf einer eID-Funktion. Üblicherweise würde eine solche Auskunft aus den folgenden Schritten bestehen:

1. Der Nutzer startet eine Auskunft, dies startet die Software für den Kartenleser.
2. Der Zweck der Auskunft und die angefragten Daten werden in der Software angezeigt. (Bzw. auf dem Display eines Komfortlesegeräts oder Standardleser mit Display.)
3. Der Nutzer überprüft die Informationen und kann, falls er möchte, die Erlaubnis für die angefragten Daten einschränken.
4. Der Nutzer autorisiert die Transaktion durch die Eingabe seines eID-PINs.
5. Der Diensteanbieter und der nPA authentifizieren sich gegenseitig.
6. Der Diensteanbieter kann nun die erlaubten Daten vom nPA auslesen.

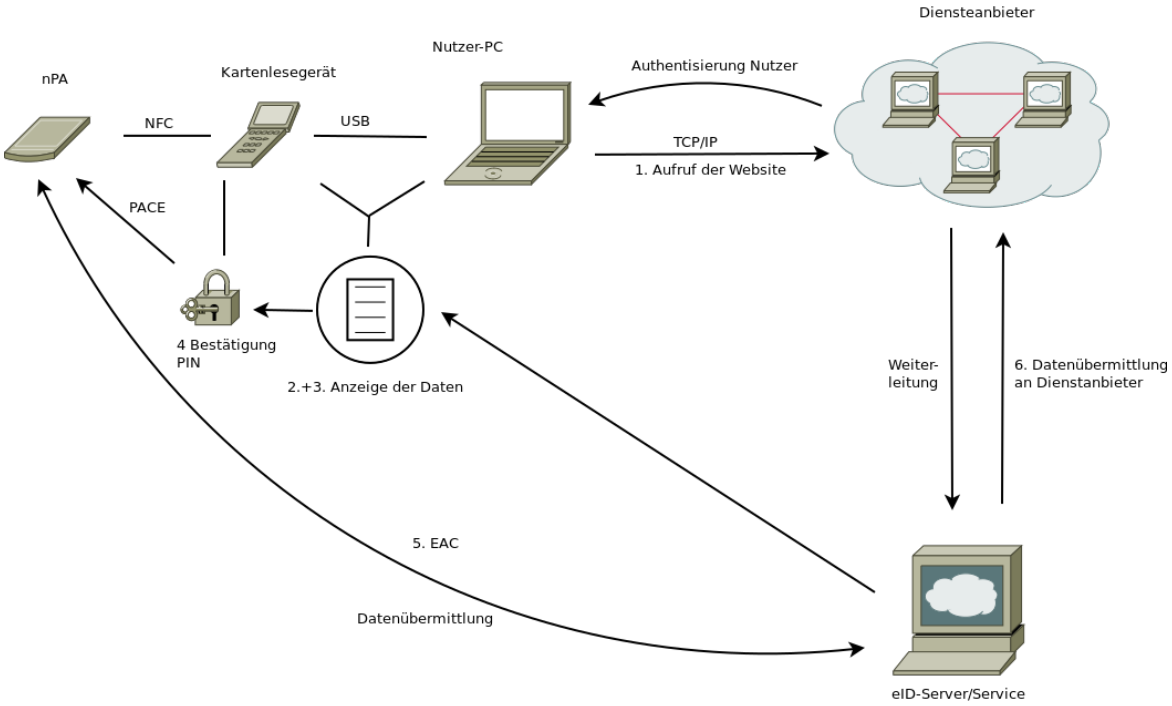

Abbildung 2: Ablauf der Kommunikation zwischen nPA und Diensteanbieter.

2.3. Password Authenticated Connection Establishment

Als Password Authenticated Connection Establishment wird der Aufbau eines passwortauthentisierten Kanals bezeichnet. Das Passwort dabei kann die CAN, die eID-PIN oder die PUK sein. Das Bundesamt für Sicherheit in der Informationstechnik (BSI) hat das Protokoll entwickelt. Die Beschreibung findet sich in der technischen Richtlinie 03110 [9]. Der dabei ausgetauschte Schlüssel hängt nicht von der Länge des verwendeten Passworts ab. Die Sicherheit des Verfahrens wurde nachgewiesen [10].

2.4. Extended Access Control

Die technische Richtlinie 03110 [9] beschreibt auch die erweiterte Zugriffskontrolle – Extended Access Control (EAC) Version 2 – für maschinenlesbare Reisedokumente. Das Protokoll EAC dient zur Absicherung sensibler Daten. Alle auf dem nPA gespeicherten personenbezogenen Daten werden als sensibel betrachtet. Die erweiterte Zugriffskontrolle besteht aus zwei Unterprotokollen: Terminal Authentication (TA) und Chip Authentication (CA), welche im Folgenden erläutert werden.

2.4.1. Terminal Authentication

Sensible Daten dürfen nur ausgelesen werden, wenn das Protokoll TA erfolgreich am Kartenlesegerät durchgeführt wurde. Dazu wird die Leseberechtigung (Terminal-Zertifikat) und das Country-Verifier-Certification-Authority-Zertifikat (CVCA-Zertifikat), sowie alle Zertifikate, die in der Zertifikatskette dazwischen stehen, an den nPA gesendet. Das CVCA-Zertifikat bildet die Wurzel für die CVCA-Public-Key-Infrastructure(PKI) und ist für den nPA vertrauenswürdig, da es schon bei der Herstellung auf dem Chip gespeichert wird.

Nachdem Echtheit und Unverfälschtheit des Terminal-Zertifikats festgestellt wurden, überprüft der nPA, ob dieses Zertifikat für diesen Dienst ausgestellt wurde. Dazu wird eine Zufallszahl an den Dienst geschickt. Diese sendet er mit seinem privaten Schlüssel signiert zurück. Mit dem öffentlichen Schlüssel aus dem Terminal-Zertifikat kann der nPA die Signatur prüfen und so feststellen, ob das Kartenlesegerät im Besitz des zum Zertifikat passenden geheimen Schlüssels ist.

2.4.2. Chip Authentication

Die Chip Authentisierung bietet eine Überprüfung der Echtheit des Chips. In jedem Chip ist ein spezielles Schlüsselpaar für asymmetrische Kryptographie gespeichert. Für anonyme Anwendungen und zum Schutz der Privatsphäre teilt sich eine Charge von Ausweisen ein Schlüsselpaar. Der private Schlüssel ist in einem bestimmten Bereich gespeichert, der nicht ausgelesen werden kann. Selbst wenn der komplette Chip kopiert werden sollte, ist es nicht möglich diesen privaten Schlüssel zu kopieren. Der Nachweis der Kenntnis des zugehörigen Schlüssels ist Beweis der Authentizität des Chips.
Zur Überprüfung sendet der Chip seinen öffentlichen Schlüssel zusammen mit einer Zufallszahl an das Kartenlesegerät. Dieses erzeugt für den Lesevorgang ein eigenes Schlüsselpaar und schickt seinen öffentlichen Schlüssel an den Chip. Beide Parteien können nun aus ihrem privaten Schlüssel, dem öffentlichen Schlüssel der anderen Partei und der Zufallszahl denselben geheimen Schlüssel berechnen. Dieser geheime und ephemere Schlüssel wird für die Absicherung der weiteren Kommunikation benutzt.

2.5. Application Protocol Data Units

Die Pakete, die zwischen der Chipkarte und der Anwendung ausgetauscht werden, werden Application Protocol Data Units (APDU) genannt. Der Befehle, der an die Chipkarte gesendet wird, wird als command APDU (cAPDU) bezeichnet, die Antwort der Karte als response APDU (rAPDU). Die Struktur der APDU ist in der ISO-Norm 7816-4 [11] spezifiziert.

2.5.1. Command APDU

Header	Body
CLA INS P1 P2	[Lc field] [Data field] [Le field]

Tabelle 1: Struktur der command APDU [11, Section 5, Figure 3]

Wie man in Tabelle 1 sieht, besteht eine cAPDU aus Header und Body. Der Header ist notwendig und beinhaltet 4 Byte: die Klasse (Class (CLA)), den Befehl (Instruction (INS)) und die beiden Parameter (P1 und P2). Der Body dagegen hat nicht immer Inhalt. Sind Daten im Body vorhanden, so ist die Länge (Lc) nicht Null und muss auch gesetzt sein. Die erwartete Länge der Antwort (Le) ist unabhängig von den Daten und muss gesetzt sein, wenn sie ungleich Null ist.
Wenn die Chipkarte sogenannte extended APDUs unterstützt, können die Längenfelder 2 Byte lang sein, sonst nur 1 Byte. Falls die Erweiterung der Längenfelder genutzt werden soll, muss vor das erste vorhandene Längenfeld der Wert 00 eingefügt werden.

2.5.2. Response APDU

Body	Trailer
[Data field]	SW1 SW2

Tabelle 2: Struktur der response APDU [11, Section 5, Figure 6]

In Abhängigkeit vom gesendeten Befehl beinhaltet die rAPDU Daten (Data field) variabler Länge gefolgt von zwei Statusbytes (SW1 und SW2). Das erste Statusbyte gibt den Status des Befehls an, das znweite kann weitere Informationen liefern.

2.5.3. Pseudo APDU

Bei den Pseudo APDUs handelt es sich um eine spezielle Klasse der APDUs. Sie sind nicht für die Chipkarte bestimmt, sondern für das Kartenlesegerät. Sie beinhalten Steuersignale für das Lesegerät.

2.6. Answer To Reset

Die Answer to Reset (ATR) ist die erste Information, die eine Chipkarte nach einem Reset sendet. Der Inhalt des ATR ist in der ISO 7816-3 [12] spezifiziert und gibt Aufschluss über die Art und Eigenschaften der Karte.

3. Methoden

3.1. Problemstellung

Inwieweit ist es möglich die eID-Funktion „in the middle“ zu nutzen? Welche Anforderungen werden dabei an den Diensteanbieter ohne eigenes Berechtigungszertifikat gestellt? Und was bedeutet die eID-Funktion „in the middle“ für den Nutzer? Könnte noch irgendjemand anders dazu in der Lage ihm dabei logisch über die Schulter zu schauen?

3.2. Lösungskonzept

Mit Diensteanbieter ist im folgenden ein Diensteanbieter ohne eigenes Berechtigungszertifikat gemeint.

Der neue Personalausweis des Nutzers wird vom Kartenlesegerät des Nutzers ausgelesen. Das Kartenlesegerät des Nutzers wird logisch mit Hilfe von Bluetooth und der Datenverbindung des Android-Telefons (UMTS/LTE oder WLAN) über die TCP/IP-Verbindung beim Rechner des Diensteanbieters eingesteckt.
Der Diensteanbieter ruft eine Selbstauskunft, wie mit einem lokalen Kartenleser, auf. Die Bestätigung der angefragten Daten sowie die eID-PIN-Eingabe erfolgen durch den Nutzer am Kartenlesegerät. Die Abbildungen 3 und 4 zeigen die entsprechende Anzeigen des Kartenlesers.
Der Prozess der Selbstauskunft bleibt durch das logische „Kabel“ bis auf eine leicht erhöhte Latenz unbeeinflusst.

In Abbildung 5 wird der geplanten Ablauf der Online-Auskunft ohne eigenes Berechtigungszertifikat gezeigt. Die Auskunft erfolgt analog zu der in Abbildung 2 dargestellten. Allerdings ruft der Diensteanbieter die Selbstauskunft auf und nicht der Nutzer. Die Aktionen und Komponenten des Nutzers sind gesondert markiert und entsprechen dem Kartenleser in der regulären Selbstauskunft.

3.3. Eingesetzte Hard- und Software

3.3.1. Android-Telefon

Das verwendete Android-Telefon muss Bluetooth LE unterstützen und mindestens Android-Version 4.4 besitzen. Hauptsächlich wurde für diese Arbeit ein *OnePlus One* verwendet. Auf diesem läuft die Android-Version 6.0.1 unter der ROM Cyanogenmod 13.0.

3.3.2. Das Kartenlesegerät

Beim verwendeten Kartenleser handelt es sich um das Modell *ReinerSCT cyberJack wave*. Dies ist ein Bluetooth-fähiger Chipkartenleser (Sicherheitsklasse 3) und unterstützt

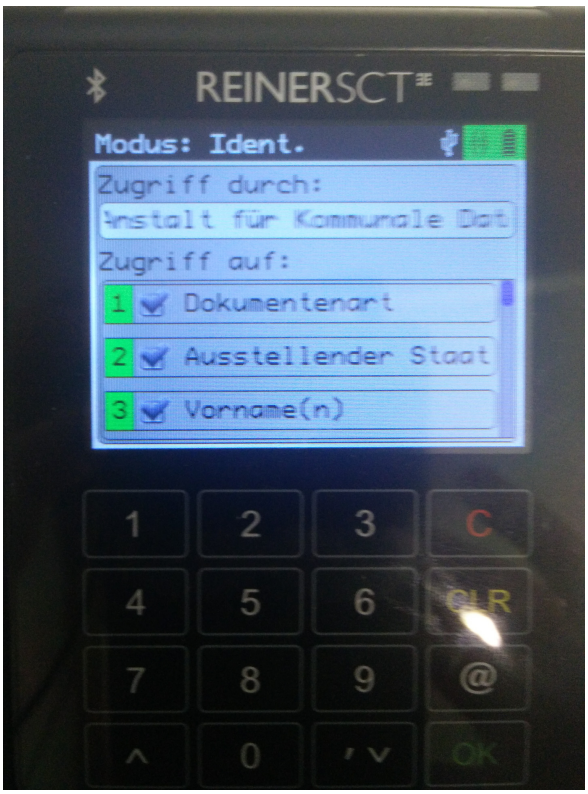

Abbildung 3: Anzeige der angefragten Daten auf dem cyberJack wave.

sowohl Bluetooth 2.1 (classic) als auch 4.0 (LE). Beim ersten Ansatz wird der Leser mit der Firmware 1.8.6 betrieben, im zweiten mit der Version 1.9.
Unter Firmwareversion 1.8.6 wurde für die Bluetooth-Kommunikation das SIM Access Profile (SAP) verwendet. Das Profil ist dazu konzipiert die Netzzugangsberechtigung einer SIM-Karte über BT weiterzuleiten. Es wird beispielsweise bei Freisprechanlagen in Kraftfahrzeugen verwendet. [13]
Nach dem Upgrade auf 1.9 wird das Generic-Attribute-Profil [14] verwendet, über das CCID [15] eingesetzt wird.

3.3.3. Logisches Kabel

Dargestellt in Abbildung 6 sind die Komponenten des logischen „Kabels“. Der erste Teil ist die TCP/IP-Verbindung zwischen Android-Telefon und Diensteanbieter. Diese könnte durch die Nutzung von stunnel [16] über SSL/TLS zusätzlich gesichert werden. Der zweite Teil die Verbindung zwischen Android-Telefon und dem Kartenlesegerät über Bluetooth LE.

3.3.4. Virtual Smart Card Architecture

Für die Verbindung zwischen Telefon und dem Server wird die *Virtual Smart Card Architecture* [17] verwendet. Serverseitig wird das Projekte *Virtual Smart Card* benutzt

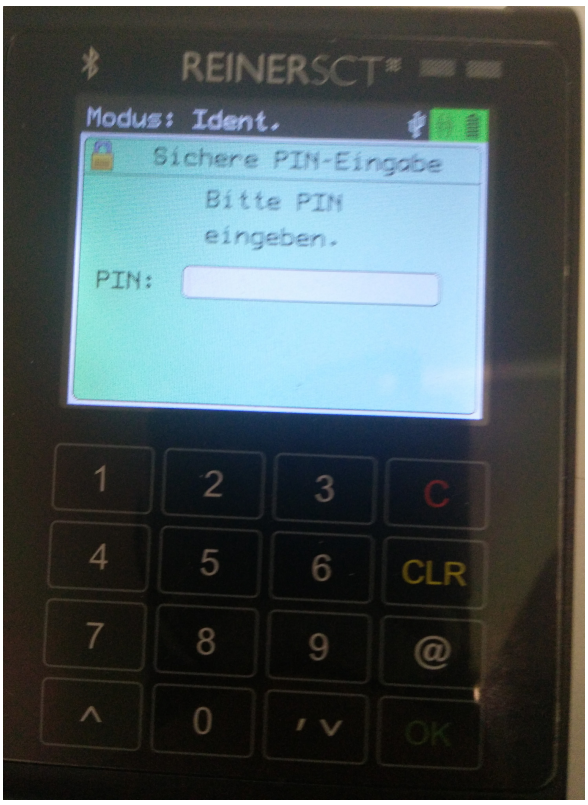

Abbildung 4: PIN-Eingabe auf dem cyberJack wave.

und die Android-App erweitert das Projekt *Remote Smart Card Reader*.
Der virtuelle Kartenleser vpcd aus dem Projekt *Virtual Smart Card* ist ein Treiber für *pcsc-lite* [18] und ermöglicht Zugriff auf virtuelle Karten. Als solche kann der weitergeleitete neue Personalausweis von einer Smart-Card-Anwendung benutzt werden.

Mit vpcd-config wird ein QR-Code erzeugt, der die Verbindungsparameter (Hostname und Port) für den *Remote Smart Card Reader* beinhaltet. Der *Remote Smart Card Reader* ist dafür gedacht NFC-fähige Android-Telefone als Kartenlesegerät zu verwenden.
Einen solchen QR-Code, wie in Abbildung 7 zu sehen, sollte der Diensteanbieter bereitstellen, damit der Nutzer die Verbindungsparameter nicht händisch eingeben muss. Da in diesem QR-Code kein Port angegeben ist, wird der Port 35963 verwendet.

3.4. Ziele der Implementierung

Das Ziel der Implementierung ist eine Android-App, die die beiden Teile des logischen „Kabels“ miteinander verbindet. Dies erfordert eine Logik für die APDUs und eine Schnittstelle zwischen dem Bluetooth-Interface und dem vpcd-worker. Wenn dies erfolgt ist, kann der verwendete Kartenleser auch entfernt benutzt werden. Dies ermöglicht einem Diensteanbieter die Selbstauskunft aufzurufen um authentisch ausgelesene Daten

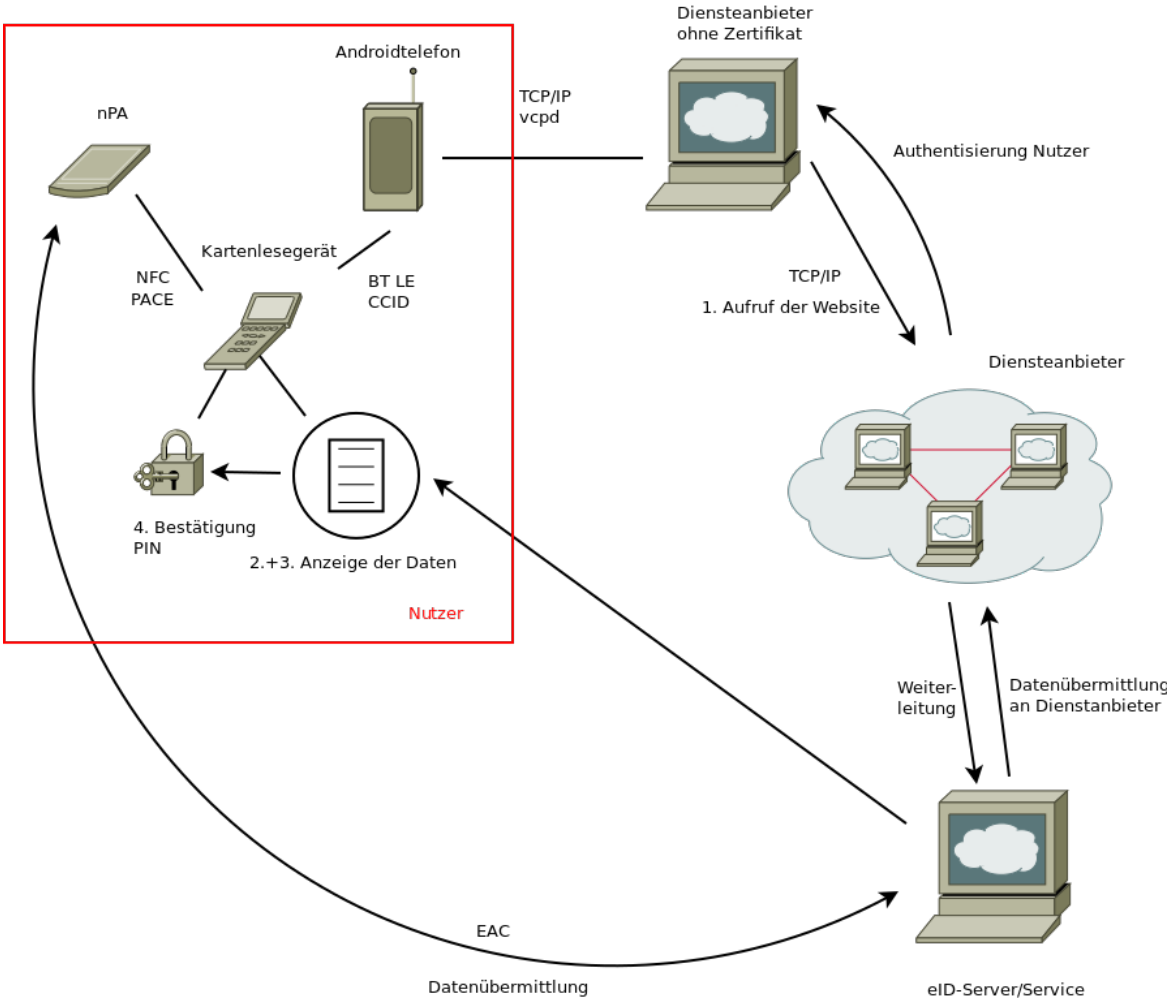

Abbildung 5: Ablauf der Kommunikation eID-Funktion „in the middle“.

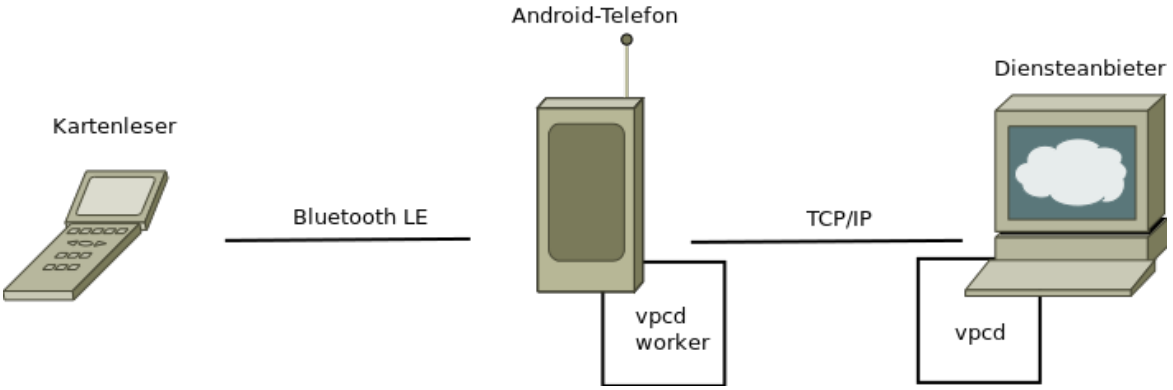

Abbildung 6: Die Komponenten des logischen „Kabels“.

vom neuen Personalausweis des Nutzers zu erhalten.

Abbildung 7: Verbindungsparameter als QR-Code

4. Ergebnisse

Dieses Kapitel ist in zwei Abschnitte unterteilt, da die zwei eingesetzten Firmwareversionen zu sehr unterschiedlichen Ergebnissen geführt haben. Der erste Ansatz wurde mit der Version 1.8.6 geführt. Der zweite Ansatz fand nach dem notwendigen Upgrade auf 1.9 statt.

4.1. Bluetooth-Kommunikation mit dem Kartenleser (1. Ansatz)

Um die eID-Funktion mit dem Kartenlesers zu nutzen wird Bluetooth LE benötigt. Dazu empfiehlt der Hersteller in den Einstellungen des Kartenlesegeräts Bluetooth 2.1 zu deaktivieren. Ansonsten koppelt sich das Android-Telefon mit dem Kartenleser im DualMode. [14]
Mit dem verwendeten *OnePlus One* war das Pairing allerdings nur im DualMode möglich. Deaktiviert man Bluetooth 2.1 am Kartenleser schlägt das Pairing fehl.

Eine mögliche Fehlerquelle ist die fehlende SAP-Unterstützung. Andere Fehlerquellen sind allerdings auch nicht auszuschließen. Die Angaben zur SAP-Unterstützung sind nur bedingt aussagekräftig, da bei SAP immer davon ausgegangen wird, dass das Mobiltelefon Zugriff auf seine SIM Karte gewährt. Mit einem *Samsung Galaxy S4*, das SAP unterstützt, traten diesbezüglich keine Fehler auf.

4.1.1. Software Development Kit

Für die Entwicklung wurde vom Hersteller Reiner SCT das Software Development Kit *RSCT_SAP_Android_V_0.9.1* bereitgestellt. Die Beispiel-App sollte eine Bluetooth-Verbindung zum Kartenlesegerät herstellen und die ATR der Karte ausgeben.

4.1.2. Transfer ATR

Die ersten Tests mit *RSCT_SAP_Android_V_0.9.1* geben allerdings nicht wie geplant die Answer to Reset (ATR) von der Karte aus, sondern folgende Antwort.

0x06	0x02	0x0000		
MsgID	Num. of Params.	Reserved		
0x02	0x00	0x0001	0x00	0x000000
ParamID	Reserved	Length	Value	Padding
0x05	0x00	0x0002	0x6700	0x0000
ParamID	Reserved	Length	Value	Padding

Tabelle 3: Response APDU

Die Nachricht ist eine *transfer_apdu_resp* und beinhaltet die Parameter *ResultCode*(0x02) mit dem Wert 0x00 (ok, request processed correctly) und *responseAPDU*(0x05) mit dem Wert 0x6700 (Länge (Lc oder Le) falsch).

Ein solcher Fehler kann an zwei Stellen entstehen, entweder wird eine fehlerhafte *command APDU* erzeugt oder das *Transport Protocol* nicht korrekt gesetzt.

Im SAP sind als mögliche Transportprotokolle T=0 und T=1 vorgesehen. Im ISO-7816 wird stattdessen der Begriff *transmission protocols* verwendet.[12]
T=0 ist das asynchrone Halbduplex Character Übertragungsprotokoll. T=1 ist das asynchrone Halbduplex Block Übertragungsprotokoll. Das sind auch die beiden Protokolle, die der nPA beherrscht.

Auch alternative *select file* Befehle (select MF) waren nicht erfolgreich.
Durch Nutzen von *ResetSIM* erhalten wir zumindest direkt eine besser aussehende Antwort. Zusätzlich musste hierfür die Funktion *sapResponseRecieved* erweitert werden, da der Service nicht die vollständige Nachricht empfangen hatte, sondern nur die ersten 19 Byte.

Header	08020000
Parameter 1	0200000100000000
Parameter 2	0600000f4b8a80018031b8738401e08290000600

Tabelle 4: Antwort mit der fehlerhaften ATR

Die Nachricht trägt den Code 0x08, es handelt sich folglich um eine *transfer_atr_resp*. Der erste Parameter ist auch hier der *ResultCode*(0x02) mit dem gewünschten Wert 0x00(ok) und der zweite ist die ATR(0x06) mit der Länge 0x000F und dem Wert `0x4B8A80018031B8738401E082900006`.
Bis auf den Initial Character stimmt die ATR. Eigentlich müsste sie mit 3b (direct convention) beginnen. Es gilt zu klären, wie dieser Fehler zu Stande kommt. Da das

gleiche Verhalten sich mit der *cyberJack Base Components* App und Bluetooth classic beobachten ließ, könnte es ein Fehler vom Lesegerät bei der Verwendung von Bluetooth sein.
Per USB wird die korrekte ATR geliefert, diese ist in Tabelle 5 dargestellt.

3B	8A	80	01	80 31 B8 73 84 01 E0 82 90 00	06
TS	T0	TD(1)	TD(2)	historical bytes	TCK

Tabelle 5: Answer To Reset

Der *initial character* TS gibt die Konvention der Karte an. Für die direkte Konvention ist der Wert auf 3B gesetzt, für die indirekte 3F.

Der *format character* T0 gibt auf den höherwertigen 4 Bit an, ob und welche *interface character* vorhanden sind. Auf den niederwertigen 4 Bit die Anzahl an *historical characters*. 8A steht folglich für das Vorhandensein von TD und 10 historischen Bytes.

Die *interface characters* TD enthalten Informationen über die möglichen Transportprotokolle. TD(1)=80 liefert in den höherwertigen 4 Bit die Information, dass TD(2) folgt und aus den niederwertigen 4 Bit Transportprotokoll T=0. TD(2)=01 bedeutet, dass keine weiteren *interface characters* folgen und Transportprotokoll T=1.

Die *historical characters* 80 31 B8 73 84 01 E0 82 90 00 beinhalten verschiedene weitere Informationen zur Karte. Das erste Byte ist der *category indicator*. [11]

Der *check character* TCK ist eine XOR-Prüfsumme von T0 bis zum letzten historischen Byte. TCK=06 ist korrekt.

4.1.3. Fazit

Mit dem verwendeten Kartenlesegerät auf Version 1.8.6 und dem SDK *RSCT_SAP_Android_V_0.9*
konnte keine sinnvolle Kommunikation mit dem neuen Personalausweis stattfinden.

4.2. Bluetooth-Kommunikation mit dem Kartenleser (2. Ansatz)

Durch das Upgrade auf die Firmwareversion 1.9 wurde der beschriebene Fehler bei der Übertragung der ATR behoben. Allerdings wurde auch die Bluetooth-Kommunikation umgestellt. Ab Version 1.9 werden zwei andere Protokolle verwendet: Secoder und CCID (chip card interface device). Ersteres wurde von der deutschen Kreditwirtschaft[3]

[3]ehemals Zentraler Kreditausschuss, seit August 2011 deutsche Kreditwirtschaft

spezifiziert und ist für den Einsatz beim Online-Banking optimiert. CCID ist der USB-Standard zur Kommunikation mit Smartcards [15].
Auch wurde mit der *reiner_ccid_via_dk_sample_0.4.1* ein neues SDK vom Hersteller für die geänderte Bluetooth-Kommunikation bereitgestellt.

4.2.1. CCID

CCID Nachrichten bestehen aus einem 10 Byte Header gefolgt von typabhängigen Daten.
Wie in Tabelle 6 zu sehen ist, unterscheidet sich die Struktur der ein- und ausgehenden Nachrichten leicht.

Bulk-IN:

bMessageTyp	dwLength	bSlot	bSeq	bStatus	bError	message specific
1	4	1	1	1	1	1

Bulk-OUT:

bMessageTyp	dwLength	bSlot	bSeq	message specific
1	4	1	1	3

Tabelle 6: CCID Header [15, S. 25]

Durch die konstante Länge des Headers beginnen die Nachrichtendaten mit einem festen Offset. Dies in Kombination mit dem Nachrichtentyp im ersten Byte ermöglicht es die Pakete schnell und einfach zu identifizieren und den Inhalt weiterzuleiten.

Wir erhalten vom *vpcd* cAPDUs. Diese müssen in einen CCID PC_to_RDR_XfrBlock umgewandelt werden. In Tabelle 7 ist die Struktur der Nachricht beschrieben. Die meisten Felder hängen von der cAPDU ab. Der Slot (bSlot) und die Wartezeit (bBWI) sind im Framework statisch gesetzt. Die Sequenznummer (bSeq) muss für jeden Block hochgezählt werden. Die Antwort enthält die Sequenznummer für die Zuordnung.

Offset	Field	Size	Value
0	bMessageType	1	0x6F
1	dwLength	4	
5	bSlot	1	0x00-FF
6	bSeq	1	0x00-FF
7	bBWI	1	0x00-FF
8	wLevelParameter	2	
10	abData	Byte array	

Tabelle 7: PC_to_RDR_XfrBlock [15, S.30]

Die Parameter(wLevelParameter) werden momentan im Framework noch nicht genutzt. Der Standard ermöglicht die Aufteilung der cAPDU auf mehrere CCID Nachrichten. Durch das verwendete Transportprotokoll T=1 und die extended APDUs ergibt sich eine maximale Blockgröße von 65544 Bytes. Solange die cAPDU dies nicht überschreitet, gibt es an dieser Stelle keinen Fehler. Die Funktion CCIDProtokoll.generateXferBlock erzeugt die korrekten Blöcke.

Die rAPDUs lassen sich wegen des oben erwähnten festen Offsets leicht extrahieren und können an den *vpcd* zurückgesendet werden.

4.2.2. Synchronisation

Während die Bluetooth LE Implementierung auf Callbacks basiert, sind für die Nutzung des vpcd-Workers lineare Funktionsaufrufe und -antworten notwendig. Dazu bietet Android die Sempaphorlösung CountDownLatch an. Für unsere Zwecke muss die Klasse noch um eine Reset-Funktion erweitert werden.

Die asynchronen Callbacks von Bluetooth LE führen auch an anderen Stellen zu Problemen. Durch die Fehleranfälligkeit der Funkverbindung kommt es häufig zu doppelt gesendeten Paketen. Durch Überprüfung der Pakete lassen sich die Dopplungen aber leicht in Software beheben.

4.3. Auskunft mit Open eCard App

Wenn zwischen der Android-App und dem Smart Card Service eine Verbindung hergestellt wurde, beginnt die *openECard App* mit der Kartenerkennung. Dazu werden einige SELECT FILE und READ BINARY Befehle an die Karte gesendet und die Ergebnisse ausgewertet. Ein Log davon befindet sich im Anhang A.1.
Nach der Erkennung ist das System bereit für eine Auskunft (siehe Abb. 8).

Abbildung 8: Screenshot Open eCard App, betriebsbereit

Wird nun eine Online-Ausweisfunktion[4] aufgerufen, reagiert die openECARD App und

[4]z.B.: https://www.buergerserviceportal.de/bund/ausweisapp/bspx_selbstauskunft

zeigt die Informationen zum Anbieter (Abb. 9) an. Bestätigt man die Anfrage, werden die angefragten Daten (Abb. 10) angezeigt. Hier hat der Nutzer die Möglichkeit Daten abzuwählen, falls die Anfrage das zulässt. Bei der Anfrage an das Kraftfahrt-Bundesamt, die in den Abbildungen als Beispiel dient, sind alle Felder benötigt und grau hinterlegt.

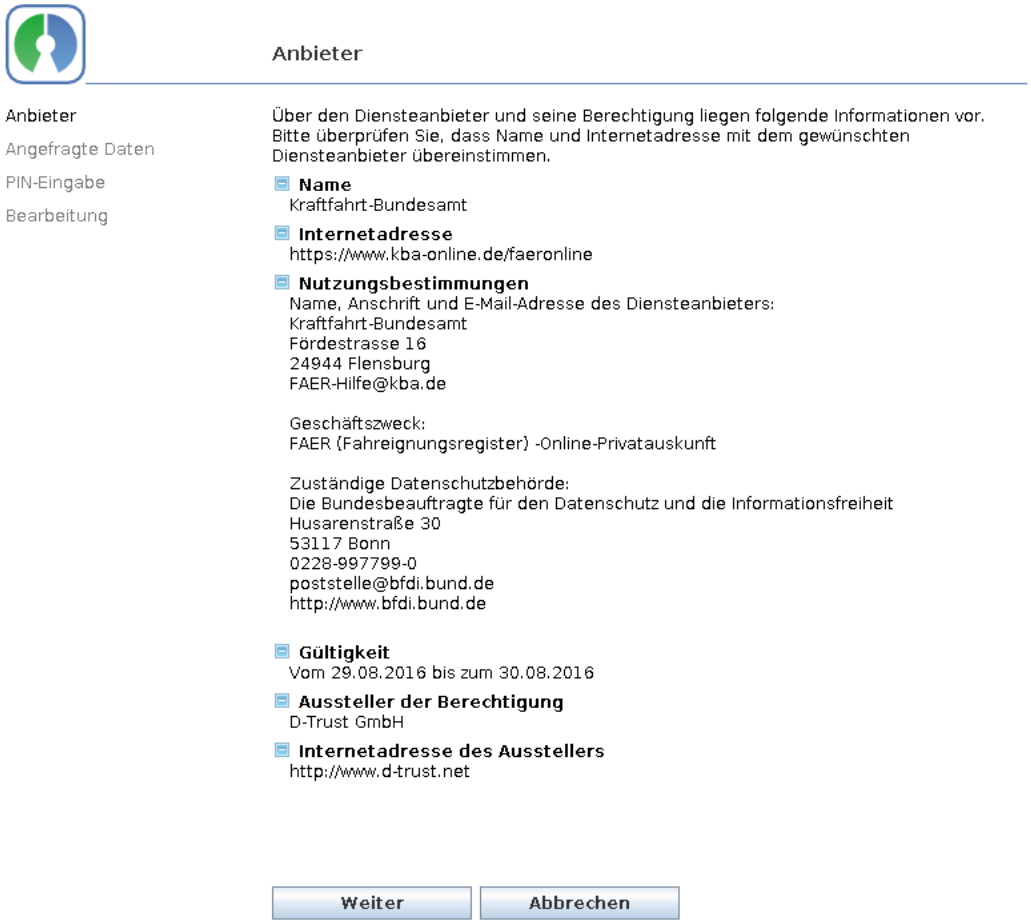

Abbildung 9: Screenshot Open eCard App, Anbieter

Als nächstes erfolgt die PIN-Eingabe. Wie man auf Abbildung 11 sehen kann, kann diese auch in der openECard App erfolgen. Das ist allerdings für unsere Zwecke nicht das gewünschte Verhalten. Grund dafür ist die Konfiguration der virtuellen Kartenleser. Wenn in dieser die Funktion PIN-Pad fehlt, wird von einem Basisleser ausgegangen, folglich muss die PIN-Eingabe am Rechner erfolgen.

Nach erfolgreicher PIN-Eingabe beginnt die gesicherte Kommunikation zwischen dem neuen Personalausweis und dem eID-Server. Ein vollständiger Log der Kommunikation befindet sich im Anhang A.2.
Im Anschluss werden die ausgelesenen Daten beim Diensteanbieter angezeigt.

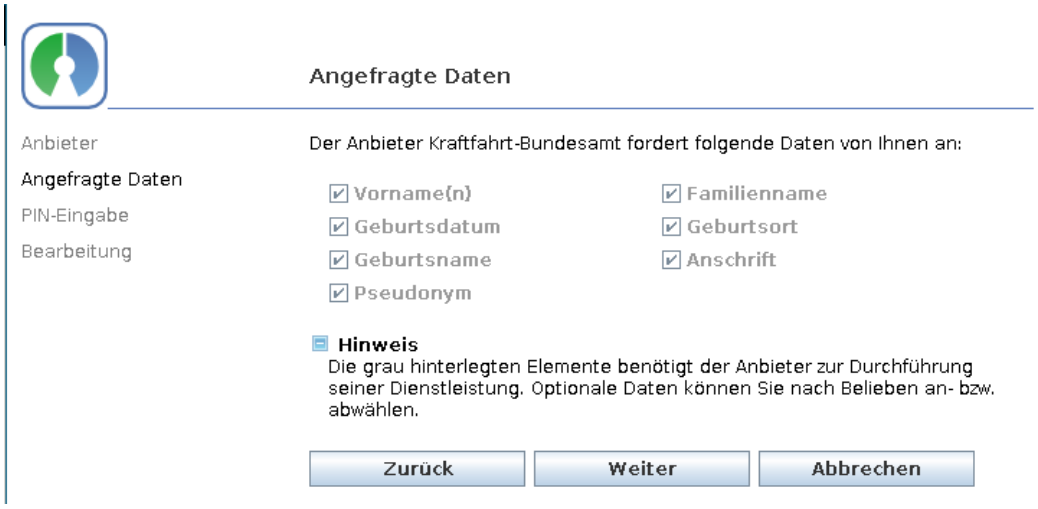

Abbildung 10: Screenshot Open eCard App, angefragte Daten

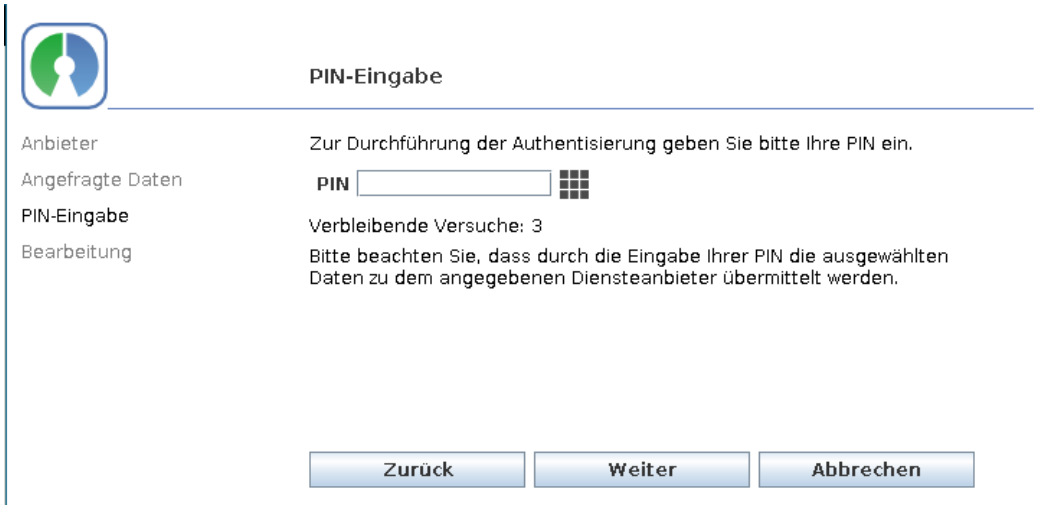

Abbildung 11: Screenshot Open eCard App, PIN-Eingabe

4.4. PIN-Eingabe

Um einen PACE-Kanal aufzubauen können Pseudo-APDUs genutzt werden. In Tabelle 8 werden zwei dafür relevante Pseudo-APDUs gezeigt.
Der erste Befehl liefert eine DER kodierte ASN.1 Struktur zurück, in der die PACE Fähigkeiten des Lesers angegeben werden. Der zweite weist das Kartenlesegerät an, einen PACE Kanal zu starten. Die Befehlsdaten sind auch eine DER kodierte ASN.1 Struktur mit den dazugehörigen Längenfeldern. Beide Befehle sind in der Technischen Richtlinie 03119 spezifiziert. [19]

Die Ausgabe von asn1parse zeigt, dass es sich um eine Sequenz handelt, in der sich ein Tag([1]) mit einem Integer(2) befindet. Ein kurzer Vergleich mit der Grammatik

	CLA	INS	P1	P2	Command Data
GetReaderPACE Capabilities	0xFF	0x9A	0x04	0x01	-
EstablishPACE Channel	0xFF	0x9A	0x04	0x02	00 00 07 30 05 A1 03 02 01 02 01 05

Tabelle 8: Beispiel Pseudo APDU

```
% openssl asn1parse -in command_data.bin -inform der -i
0:d=0  hl=2 l=   5 cons: SEQUENCE
2:d=1  hl=2 l=   3 cons:  cont [ 1 ]
4:d=2  hl=2 l=   1 prim:   INTEGER           :02
```

Tabelle 9: Ausgabe von asn1parse

für EstablishPACEChannelInput aus der TR-03119 offenbart, dass es sich damit um die kürzeste valide Sequenz handelt. Der mit passwordID bezeichnete Integer darf die Werte 2(CAN), 3(PIN) und 4(PUK) annehmen.

```
EstablishPACEChannelInput ::= SEQUENCE {
    passwordID              [1] INTEGER
    transmittedPassword     [2] NumericString OPTIONAL
    cHAT                    [3] OCTET STRING OPTIONAL
    -- as defined in [TR-03110], including tag 0x7F4C
    certificateDescription  [4] CertificateDescription OPTIONAL
    hashOID                 [5] OBJECT IDENTIFIER OPTIONAL
}
```

Tabelle 10: Definition EstablishPACEChannelInput [19, S. 33f]

In der aktuellen Firmwareversion 1.9 führt diese APDU bei dem von uns verwendeten Kartenlesegerät noch zum Absturz des Kartenlesegeräts.
Es konnte verifiziert werden, dass diese Pseudo APDU auch wenn der Kartenleser per USB angeschlossen ist, zum Absturzes des verwendeten Kartenlesegeräts führt.

Unter der Verwendung des *reiner_ccid_via_dk_sample_0.4.1* und dem Kartenleser mit Firmwareversion 1.9 lässt sich die Verbindung zwischen nPA und *vpcd* herstellen. Die Kartenerkennung funktioniert, d.h. die Daten werden auch korrekt übertragen. Der Aufbau von PACE ist noch nicht fehlerfrei.

5. Diskussion

Im Rahmen dieser Bachelorarbeit wurde eine Software konzipiert und implementiert, die es dem Nutzer ermöglicht sein Android-Telefon über Bluetooth mit einem *cyberJack wave* zu verbinden und basierend auf der *Virtual Smart Card Architecure* an einem entfernten Server anzuschließen. Damit können APDUs zwischen einer Anwendung auf diesem Server und dem neuen Personalausweis im Kartenlesegerät gesendet und beantwortet werden. Dies ist die Voraussetzung um die eID-„in-the-middle“ nutzen zu können.

In diesem Kapitel wird ersten Abschnitt auf die Auswertung der Ergebnisse eingegangen. Danach folgt in den Abschnitten 5.2 und 5.3 die Implikationen der Ergebnisse für den Diensteanbieter bzw. Nutzer ein. Im Anschluss wird in Abschnitt 5.4 ein Ausblick auf mögliche Erweiterungen und Verbesserungen der Software gegeben.

5.1. Auswertung

Wie in Kapitel 4 gezeigt wurde, ist es grundsätzlich möglich eine Verbindung zwischen dem nPA und dem *vpcd* herzustellen. Da die APDUs auf dem Telefon verarbeitet werden, ist es möglich sein die Pseudo APDU abzufangen und stattdessen die CCID Variante des Befehls (PC_to_RDR_Secure) zu nutzen. Dies zu testen war im zeitlichen Rahmen dieser Arbeit nicht möglich.
Ein limitierender Faktor bei der Implementierung lag in der eingesetzten Hardware. Bei den zwei Funkverbindungen NFC und Bluetooth kommt es häufig zu Fehlern oder Dopplungen bei der Übertragung. Dies erschwerte die Suche nach Fehlerursachen.

5.2. Situation für den Diensteanbieter

Es ist für den Diensteanbieter möglich eID-„in-the-middle“ zu nutzen. Folglich kann er auch ohne eigenes Berechtigungszertifikat und eID-Server authentische ausgelesene Daten vom neuen Personalausweises des Nutzers erhalten. Die Daten sind durch die reguläre Selbstauskunft transportverschlüsselt.

5.3. Situation für den Nutzer

Für den Nutzer entsteht durch eID-„in-the-middle“ kein Nachteil. Das Kartenlesegerät zeigt sicher an, durch wen der Zugriff auf den neuen Personalausweis erfolgt und welche Datengruppen dabei ausgelesen werden (Siehe Abbildung 3). Auch die PIN-Eingabe ist sicher und direkt am Kartenlesegerät. Wenn die in Kapitel 3 vorgeschlagene Sicherung der Verbindung zwischen *vpcd* und Telefon umgesetzt ist, sind alle Strecken transportverschlüsselt. Folglich muss der Nutzer sich auch keine Sorgen machen, dass ihm jemand anders über die Schulter schaut. Allerdings muss der Diensteanbieter nicht die Bedingungen für das Berechtigungszertifikat erfüllen. Diese beinhalten unter

anderem auch Maßnahmen zu Datenschutz und Datensicherheit. Das Vertrauen in den Diensteanbieter muss der Nutzer allerdings in jedem Fall erbringen.

5.4. Ausblick

Das Ziel wäre eID-„in-the-middle“ so anzubieten, dass sich der Nutzer lediglich die Android-App installieren muss, den QR-Code des Diensteanbieters scannt und sich so authentifizieren kann. Dazu wäre es sinnvoll die Benutzerschnittstelle, die Fehlererkennung und Stabilität der Android-App zu verbessern
Um dies praktisch einsetzten zu können, wird noch ein automatisierter Aufruf der Selbstauskunft benötigt. Außerdem muss einer Web-Anfrage noch der korrekte virtuelle Kartenleser, d.h. der Port in den Verbindungsparametern, zugewiesen werden.
Innerhalb der *Virtual Smart Card Architecture* könnte man sicherstellen, dass das entfernte Kartenlesegerät mit den korrekten Eigenschaften erkannt wird. Wenn das PIN-PAD nicht erkannt wird, muss davon ausgegangen werden, dass der Kartenleser dies nicht unterstützt.

A. Anhang

A.1. Log: Kartenerkennung

```
Connecting to 141.20.9.17:35963...
Connected to VPCD
Powered up the card with ATR 3B8A80018031B8738401E082900006
C-APDU: 00A4000C023F00
R-APDU: 9000
C-APDU: 00A4020C020003
R-APDU: 6A82
C-APDU: 00A4000C023F00
R-APDU: 9000
C-APDU: 00A4020C022F00
R-APDU: 9000
C-APDU: 00B20404FF
R-APDU: 6D00
C-APDU: 00A4000C023F00
R-APDU: 9000
C-APDU: 00A4020C022F02
R-APDU: 6A82
C-APDU: 00A4000C023F00
R-APDU: 9000
C-APDU: 00A4020C022F00
R-APDU: 9000
C-APDU: 00B20304FF
R-APDU: 6D00
C-APDU: 00A4040C0FF045737445494420766572203 12E30
R-APDU: 6A82
C-APDU: 00A4040C0FD233000000457374454944207633 35
R-APDU: 6A82
C-APDU: 00A4040008276000121F000001
R-APDU: 6A82
C-APDU: 00A4000C023F00
R-APDU: 9000
C-APDU: 00A4020C022F00
R-APDU: 9000
C-APDU: 00B00000FF
R-APDU: 61324F0FE828BD080FA000000167455349474E500F434941207A752044
462E655369676E5100730C4F0AA000000167455349474E61094F07A00000024710
01610B4F09E80704007F00070302610C4F0AA000000167455349474E6282
```

A.2. Log: Selbstauskunft

```
00000000 APDU: 00 A4 00 0C 02 3F 00
00069049 SW: 90 00
00000901 APDU: 00 A4 02 0C 02 00 03
00006059 SW: 6A 82
00000796 APDU: 00 A4 00 0C 02 3F 00
00007198 SW: 90 00
00000679 APDU: 00 A4 02 0C 02 2F 00
00008345 SW: 90 00
00001838 APDU: 00 B2 04 04 FF
00004140 SW: 6D 00
00003222 APDU: 00 A4 00 0C 02 3F 00
00015107 SW: 90 00
00004242 APDU: 00 A4 02 0C 02 2F 02
00006430 SW: 6A 82
00000749 APDU: 00 A4 00 0C 02 3F 00
00012065 SW: 90 00
00001877 APDU: 00 A4 02 0C 02 2F 00
00009307 SW: 90 00
00000837 APDU: 00 B2 03 04 FF
00004227 SW: 6D 00
00001097 APDU: 00 A4 04 0C 0F F0 45 73 74 45 49 44 20 76 65 72 20 31 2E 30
00013855 SW: 6A 82
00001005 APDU: 00 A4 04 0C 0F D2 33 00 00 00 45 73 74 45 49 44 20 76 33 35
00009998 SW: 6A 82
00000633 APDU: 00 A4 04 00 08 27 60 00 12 1F 00 00 01
00009353 SW: 6A 82
00000628 APDU: 00 A4 00 0C 02 3F 00
00010371 SW: 90 00
00000815 APDU: 00 A4 02 0C 02 2F 00
00009181 SW: 90 00
00000580 APDU: 00 B0 00 00 FF
00010429 SW: 61 32 4F 0F E8 28 BD 08 0F A0 00 00 01 67 45 53 49 47 4E 50 0F 43 49 41 20 7A
    75 20 44 46 2E 65 53 69 67 6E 51 00 73 0C 4F 0A A0 00 00 01 67 45 53 49 47 4E 61 09
    4F 07 A0 00 00 02 47 10 01 61 0B 4F 09 E8 07 04 00 7F 00 07 03 02 61 0C 4F 0A A0 00
    00 01 67 45 53 49 47 4E 62 82
30858580 APDU: 00 A4 00 0C 02 3F 00
00008185 SW: 90 00
00635942 APDU: 00 22 C1 A4 0F 80 0A 04 00 7F 00 07 02 02 04 02 02 83 01 03
00070028 SW: 90 00
27107505 APDU: 00 84 00 00 08
00037295 SW: 97 77 14 46 20 F9 6B 47 90 00
00406990 APDU: 00 22 81 B6 10 83 0E 44 45 43 56 43 41 65 49 44 30 30 31 30 34
00056016 SW: 90 00
00002387 APDU: 00 2A 00 BE E7 7F 4E 81 A0 5F 29 01 00 42 0E 44 45 43 56 43 41 65 49 44 30
    30 31 30 34 7F 49 4F 06 0A 04 00 7F 00 07 02 02 02 02 03 86 41 04 0C E7 A1 33 AC 9D
    24 1B CB B0 18 E2 43 0A C0 EA F4 63 A4 72 B3 9B 18 05 FD CC 4E 5C 99 0C 90 98 98 71
    9D B9 0D 29 14 7B 13 D5 B7 7F 33 E9 EC 2A 9F D3 40 B6 FA 19 98 71 8D F6 5A 49 EC 1C
    B7 61 5F 20 10 44 45 44 56 65 49 44 44 54 52 31 30 31 34 32 34 7F 4C 12 06 09 04 00
    7F 00 07 03 01 02 02 53 05 40 05 13 FF 87 5F 25 06 01 06 00 06 01 07 5F 24 06 01 06
    00 09 01 05 5F 37 40 A5 13 63 00 F0 62 72 AB 1B A6 85 B7 F5 86 48 EE BD 11 68 67 05
    E1 4D 33 6B AA 56 C3 BF F4 90 74 66 26 0B 58 EA DB CD E3 A7 A8 E1 50 45 BB 8B 0A 51
    95 E1 08 9E ED AA 81 DF E5 F8 07 03 9F 12 7C
00333610 SW: 90 00
00001032 APDU: 00 22 81 B6 12 83 10 44 45 44 56 65 49 44 44 54 52 31 30 31 34 32 34
00050931 SW: 90 00
00002416 APDU: 00 2A 00 BE 00 01 48 7F 4E 82 01 00 5F 29 01 00 42 10 44 45 44 56 65 49 44
    44 54 52 31 30 31 34 32 34 7F 49 4F 06 0A 04 00 7F 00 07 02 02 02 02 03 86 41 04 05
    D3 60 04 16 0B A1 25 08 14 1A E0 19 CF 01 21 6B C4 E4 57 FB 9D A8 9F B3 FD F1 FD 64
    22 94 D9 A7 7D A2 A6 60 05 C3 D9 54 43 14 6F D9 E7 99 C2 0C CD 7B 12 B0 C3 C3 BC EC
    E9 9D 64 BF 29 CA B0 5F 20 0E 44 45 30 30 30 30 33 35 36 30 30 35 38 36 7F 4C 12 06
    09 04 00 7F 00 07 03 01 02 02 53 05 00 05 13 FB 07 5F 25 06 01 06 00 09 01 02 5F 24
    06 01 06 00 09 01 03 65 5E 73 2D 06 09 04 00 7F 00 07 03 01 03 02 80 20 6B A1 D8 DE
    B2 82 F4 22 40 97 9B F4 37 81 26 99 4A F9 34 73 DF F5 DB A8 59 BB DE 56 2A B2 99 55
    73 2D 06 09 04 00 7F 00 07 03 01 03 01 80 20 25 8B 9A 69 B9 33 DF EE CA 38 F1 40 55
```

```
    20 F5 3D 46 A6 5A 26 26 BC 12 5E 56 DA 14 5D D8 B8 E4 C3 5F 37 40 16 11 45 7B BE B3
    83 DB E9 E3 46 88 57 C9 91 6F 39 13 C6 0B 1A 44 90 EB A4 A4 33 4F 1F 5C B2 50 4A 84
    32 65 47 BC E6 CE 66 F5 8A DF 74 BC DA D8 1B 6C 3E 17 BA 1B A8 DD CD 98 AF 48 DF A8
    A7 D2
00321613 SW: 90 00
00040963 APDU: 00 22 81 A4 57 80 0A 04 00 7F 00 07 02 02 02 02 03 83 0E 44 45 30 30 30 30
    33 35 36 30 30 35 38 36 91 20 2C 11 C9 A6 DA 6D 3C B5 08 6A FC D6 16 33 31 52 63 A8
    3D C1 F4 B8 4E 9C 98 28 16 99 6A EB 3C E7 67 17 73 15 06 09 04 00 7F 00 07 03 01 04
    02 53 08 32 30 31 36 30 39 31 32
00068020 SW: 90 00
00001220 APDU: 00 82 00 00 40 75 81 37 3B 28 A8 5F 4C EF 25 C5 1E DB 7C F7 DE 1B 1D 3D 32
    60 08 7F 28 59 00 99 B5 3B 27 58 5A 82 67 23 05 F5 17 46 5B 06 48 8C 0E 0C AD 77 9B
    1A 32 78 04 B3 F2 46 5C 08 B8 38 8D 71 F2 BC 24
00231770 SW: 90 00
00002713 APDU: 00 A4 02 04 02 01 1D FF
00041284 SW: 62 1A 80 02 09 00 C5 02 06 61 82 01 01 83 02 01 1D 88 01 E8 8A 01 05 A1 03 8B
    01 04 90 00
00003011 APDU: 00 B0 00 00 FF
00037027 SW: 30 82 06 5D 06 09 2A 86 48 86 F7 0D 01 07 02 A0 82 06 4E 30 82 06 4A 02 01 03
    31 0F 30 0D 06 09 60 86 48 01 65 03 04 02 01 05 00 30 82 01 48 06 08 04 00 7F 00 07
    03 02 01 A0 82 01 3A 04 82 01 36 31 82 01 32 30 0D 06 08 04 00 7F 00 07 02 02 02 02
    01 02 30 12 06 0A 04 00 7F 00 07 02 02 03 02 02 02 01 02 02 01 48 30 12 06 0A 04 00
    7F 00 07 02 02 04 02 02 02 01 02 02 01 0D 30 17 06 0A 04 00 7F 00 07 02 02 05 02 03
    30 09 02 01 01 02 01 4A 01 01 FF 30 17 06 0A 04 00 7F 00 07 02 02 05 02 03 30 09 02
    01 01 02 01 4C 01 01 00 30 19 06 09 04 00 7F 00 07 02 02 05 02 30 0C 06 07 04 00 7F
    00 07 01 02 02 01 0D 30 1C 06 09 04 00 7F 00 07 02 02 03 02 30 0C 06 07 04 00 7F 00
    07 01 02 02 01 0D 02 01 48 30 2A 06 08 04 00 7F 00 07 02 02 06 16 1E 68 74 74 70 3A
    2F 2F 62 73 69 90 00
00002047 APDU: 00 B0 00 FF FF
00036959 SW: 2E 62 75 6E 64 2E 64 65 2F 63 69 66 2F 6E 70 61 2E 78 6D 6C 30 62 06 09 04 00
    7F 00 07 02 02 01 02 30 52 30 0C 06 07 04 00 7F 00 07 01 02 02 01 0D 03 42 00 04 3A
    63 01 5A 9C 74 D8 92 3C FA 66 5C 53 90 48 5D 42 05 F5 99 31 B1 5B AF D1 43 AF AE 23
    56 5C CA 9E 59 F5 83 A4 7D 76 16 89 4A 2E BA F8 46 99 65 43 92 1E DD A9 26 35 D3 2E
    B1 50 60 6F 30 6D 41 02 01 48 A0 82 03 D3 30 82 03 CF 30 82 03 56 A0 03 02 01 02 02
    01 2A 30 0A 06 08 2A 86 48 CE 3D 04 03 03 30 4F 31 0B 30 09 06 03 55 04 06 13 02 44
    45 31 0D 30 0B 06 03 55 04 0A 0C 04 62 75 6E 64 31 0C 30 0A 06 03 55 04 0B 0C 03 62
    73 69 31 0C 30 0A 06 03 55 04 05 13 03 31 30 30 31 15 30 13 06 03 55 04 03 0C 0C 63
    73 63 61 2D 67 65 72 6D 61 6E 79 30 1E 17 0D 31 32 30 35 32 33 31 33 34 34 31 36 5A
    17 0D 32 32 31 90 00
00001863 APDU: 00 B0 01 FE FF
00037121 SW: 31 32 33 32 33 35 39 35 39 5A 30 67 31 0B 30 09 06 03 55 04 06 13 02 44 45 31
    1D 30 1B 06 03 55 04 0A 0C 14 42 75 6E 64 65 73 64 72 75 63 6B 65 72 65 69 20 47 6D
    62 48 31 0C 30 0A 06 03 55 04 05 13 03 31 31 37 31 2B 30 29 06 03 55 04 03 0C 22 44
    6F 63 75 6D 65 6E 74 20 53 69 67 6E 65 72 20 49 64 65 6E 74 69 74 79 20 44 6F 63 75
    6D 65 6E 74 73 30 82 01 33 30 81 EC 06 07 2A 86 48 CE 3D 02 01 30 81 E0 02 01 01 30
    2C 06 07 2A 86 48 CE 3D 01 01 02 21 00 A9 FB 57 DB A1 EE A9 BC 3E 66 0A 90 9D 83 8D
    72 6E 3B F6 23 D5 26 20 28 20 13 48 1D 1F 6E 53 77 30 44 04 20 7D 5A 09 75 FC 2C 30
    57 EE F6 75 30 41 7A FF E7 FB 80 55 C1 26 DC 5C 6C E9 4A 4B 44 F3 30 B5 D9 04 20 26
    DC 5C 6C E9 4A 4B 44 F3 30 B5 D9 BB D7 7C BF 95 84 16 29 5C F7 E1 CE 6B CC DC 18 FF
    8C 07 B6 04 41 90 00
00019566 APDU: 00 B0 02 FD FF
00037439 SW: 04 8B D2 AE B9 CB 7E 57 CB 2C 4B 48 2F FC 81 B7 AF B9 DE 27 E1 E3 BD 23 C2 3A
    44 53 BD 9A CE 32 62 54 7E F8 35 C3 DA C4 FD 97 F8 46 1A 14 61 1D C9 C2 77 45 13 2D
    ED 8E 54 5C 1D 54 C7 2F 04 69 97 02 21 00 A9 FB 57 DB A1 EE A9 BC 3E 66 0A 90 9D 83
    8D 71 8C 39 7A A3 B5 61 A6 F7 90 1E 0E 82 97 48 56 A7 02 01 01 03 42 00 04 03 60 DA
    23 38 16 AD A8 5C 20 07 39 12 F6 43 B9 76 0F 6E 30 87 AD 97 4B 05 0A 5A DC 3B FF 88
    A8 36 59 70 3A DA C3 FF 51 B7 1F 97 6C B2 7E 89 34 15 9C 42 98 57 BE 7A A6 E7 C4 65
    90 49 F3 74 1F A3 82 01 2D 30 82 01 29 30 1F 06 03 55 1D 23 04 18 30 16 80 14 E3 76
    AE 66 12 FE 7A 81 E6 72 2C 51 38 5B D8 83 49 0F C3 A2 30 1D 06 03 55 1D 0E 04 16 04
    14 71 53 85 28 24 36 FE 7A AD 6F 83 93 E1 AB 4D 1C A1 7E B8 02 30 0E 06 03 55 1D 0F
    01 01 FF 04 04 90 00
00002129 APDU: 00 B0 03 FC FF
00036867 SW: 03 02 07 80 30 2B 06 03 55 1D 10 04 24 30 22 80 0F 32 30 31 32 30 35 32 33 31
    33 34 34 31 36 5A 81 0F 32 30 31 33 30 31 32 33 32 33 35 39 35 39 5A 30 16 06 03 55
    1D 20 04 0F 30 0D 30 0B 06 09 04 00 7F 00 07 03 01 01 01 30 1D 06 03 55 1D 11 04 16
    30 14 82 12 62 75 6E 64 65 73 64 72 75 63 6B 65 72 65 69 2E 64 65 30 41 06 03 55 1D
```

```
    12 04 3A 30 38 81 18 63 73 63 61 2D 67 65 72 6D 61 6E 79 40 62 73 69 2E 62 75 6E 64
    2E 64 65 86 1C 68 74 74 70 73 3A 2F 2F 77 77 77 2E 62 73 69 2E 62 75 6E 64 2E 64 65
    2F 63 73 63 61 30 30 06 03 55 1D 1F 04 29 30 27 30 25 A0 23 A0 21 86 1F 68 74 74 70
    3A 2F 2F 77 77 77 2E 62 73 69 2E 62 75 6E 64 2E 64 65 2F 63 73 63 61 5F 63 72 6C 30
    0A 06 08 2A 86 48 CE 3D 04 03 03 03 67 00 30 64 02 30 13 B1 07 4B 3A F6 38 99 5C 15
    B1 D8 FD D8 7F 90 00
00002179 APDU: 00 B0 04 FB FF
00036826 SW: 1F 36 16 E1 95 28 C5 25 AE A5 F3 A1 E6 4B 31 FE 8F 30 CA 14 E3 A0 41 07 52 12
    C8 2A 86 A8 B8 38 2D 02 30 4E F1 B8 F4 09 C3 76 61 B6 74 3A DC E7 C3 6D B0 25 F7 FF
    74 1F BF 20 16 44 A8 66 A1 86 A7 96 01 ED 76 56 85 8F B0 8F 62 A4 9E 09 85 79 8A AE
    F2 31 82 01 0F 30 82 01 0B 02 01 01 30 54 30 4F 31 0B 30 09 06 03 55 04 06 13 02 44
    45 31 0D 30 0B 06 03 55 04 0A 0C 04 62 75 6E 64 31 0C 30 0A 06 03 55 04 0B 0C 03 62
    73 69 31 0C 30 0A 06 03 55 04 05 13 03 31 30 30 31 15 30 13 06 03 55 04 03 0C 0C 63
    73 63 61 2D 67 65 72 6D 61 6E 79 02 01 2A 30 0D 06 09 60 86 48 01 65 03 04 02 01 05
    00 A0 4A 30 17 06 09 2A 86 48 86 F7 0D 01 09 03 31 0A 06 08 04 00 7F 00 07 03 02 01
    30 2F 06 09 2A 86 48 86 F7 0D 01 09 04 31 22 04 20 B3 21 89 63 0B 7D BE 45 6A 4C 95
    D2 6E 1F 52 01 90 00
00001624 APDU: 00 B0 05 FA FF
00030359 SW: 55 F1 F0 5E 4E E0 64 74 98 87 63 F8 50 92 80 62 30 0C 06 08 2A 86 48 CE 3D 04
    03 02 05 00 04 47 30 45 02 21 00 8A 6B 6A 6F 12 65 C7 E3 35 35 13 CC 2F B7 FF 1E CA
    E6 93 2E 67 54 B6 90 F4 AF 25 2B 3C 0F 7B 06 02 20 28 45 1E DD 08 75 32 6F AD 91 17
    93 BB F4 11 A9 AF 43 53 92 C7 8C BF 1A 56 8C E6 2C F8 A2 0D 7F 62 82
00001217 APDU: 00 22 41 A4 0F 80 0A 04 00 7F 00 07 02 02 03 02 02 84 01 48
00046759 SW: 90 00
00001071 APDU: 00 86 00 00 45 7C 43 80 41 04 2C 11 C9 A6 DA 6D 3C B5 08 6A FC D6 16 33 31
    52 63 A8 3D C1 F4 B8 4E 9C 98 28 16 99 6A EB 3C E7 A1 B9 A0 DF 4B 4C 09 F5 BF EE E2
    6B 4C BC 2B 9E 45 07 0A 2F E6 45 AF 02 26 B3 33 74 00 C0 0C C4 00
00202971 SW: 7C 14 81 08 59 CF 45 2C C6 E6 00 41 82 08 DF C5 D1 95 32 1B 3D EC 90 00
00342428 APDU: 0C A4 04 0C 1D 87 11 01 CF BB F0 9D C2 36 42 EA 3F 05 AD 2F FF D2 61 E4 8E
    08 D9 C1 7D 30 A6 EC AE 97 00
00080532 SW: 99 02 90 00 8E 08 79 D7 62 7F 6E 8B B3 CA 90 00
00000698 APDU: 8C 20 80 00 1D 87 11 01 AA F2 52 BF 41 A0 29 6E 13 A0 0A 01 36 86 37 D4 8E
    08 AC 43 F3 CE C5 37 B4 96 00
00048298 SW: 99 02 90 00 8E 08 40 57 A0 A5 97 0B 84 CB 90 00
00000511 APDU: 0C 22 41 A4 1D 87 11 01 94 E9 AC D0 F2 D0 3D F1 52 9E EA 53 42 52 9A 50 8E
    08 BC 3D E2 AD DB 91 5F DD 00
00130511 SW: 99 02 90 00 8E 08 C3 BF B5 57 76 DF 68 BC 90 00
00001120 APDU: 0C 86 00 00 00 01 42 87 82 01 31 01 26 1C 6F F4 2E 31 97 8E F0 88 85 63 AF
    24 41 36 4B 1A 4C C8 4F EB AC DC 6B E8 0E E0 1B A0 EA 67 C4 A2 1C 0C 59 5A 5A 84 A4
    4A D0 05 87 05 A0 3C 02 A7 C6 6E 22 AB 4F 02 33 AD 19 B8 4A 5C 6B 69 FE 6D 7A 4D C7
    92 32 B2 72 06 43 46 AF 2D 9C A7 97 43 BA 37 0A E6 17 6F 89 C6 17 41 E6 A4 9D 4B E1
    8D 63 CD 6B 94 58 8C 51 14 99 D4 F2 24 DF 45 23 BF B0 CC DE 38 AD 6D 69 F3 52 E3 FC
    A1 EA FC 7B 92 A0 2C 8C 94 13 29 A8 96 2D F0 B5 B9 47 81 5B 12 C8 FB 5F F2 3E 0B ED
    B6 CF A0 7C 5B B5 8E 49 91 E9 55 67 72 9E 62 D5 BE A3 A7 A5 1C 12 BC FC 24 A8 3B 1B
    2E 23 73 90 97 6B 01 F1 AC BA CA 61 4A FA F8 B9 0B 37 1B AD 71 9B 5C 1C 00 6E 93 E8
    E8 E8 B4 3D 9C 21 FB C1 BC 7B 47 86 EF 56 CD C3 BA 42 E6 50 75 D3 4F AA FD E3 92 B2
    75 4E 31 5C 85 20 12 16 4D 49 BB 26 89 C4 56 FB 3E D1 04 B5 79 14 9B F8 0E 36 61 56
    3B D5 8C 95 34 57 B2 29 19 30 15 2A 9C 4A 34 F5 4E B2 51 38 88 F3 F1 47 77 D8 64 ED
    2C B0 6A 32 61 6A 5B 14 F0 BE 52 97 01 24 8E 08 F2 63 5D 0D 42 03 5C 3E 00 00
00249880 SW: 87 31 01 51 8A DB 0A AA 04 46 7C 38 29 9F 67 DE 6A ED F9 99 2E F6 D1 B3 C0 4B
    FD 6D C2 6C AE 39 05 FD B3 33 14 AA 70 46 8E 20 DC 47 DD D4 E8 40 0D 41 A5 99 02 90
    00 8E 08 19 8E 35 06 28 31 73 67 90 00
00263644 APDU: 0C B0 81 00 00 00 0E 97 02 00 00 8E 08 1A 2B 92 FD 29 DA 8A EB 00 00
00105352 SW: 87 11 01 AC 91 AD 95 BF C1 03 8E BB 2F 0B E8 7E C4 98 23 99 02 90 00 8E 08 A9
    93 F6 75 76 D9 87 15 90 00
00000695 APDU: 0C B0 82 00 00 00 0E 97 02 00 00 8E 08 6D 46 0D 33 C6 50 AA 7F 00 00
00097282 SW: 87 11 01 07 72 4C BF 9F 1C CC 43 59 7F 2D E6 DB 9A 26 A3 99 02 90 00 8E 08 A0
    B7 B9 F2 0B D4 18 F9 90 00
00000543 APDU: 0C B0 84 00 00 00 0E 97 02 00 00 8E 08 D8 76 0C 5D 02 BC 9E 3F 00 00
00107486 SW: 87 11 01 8F E9 85 00 82 6C AD 1A B6 5E CE 66 9D FE 03 F6 99 02 90 00 8E 08 D8
    98 E3 C1 93 0D 2E D5 90 00
00000760 APDU: 0C B0 85 00 00 00 0E 97 02 00 00 8E 08 E8 86 B6 AD FF 2C 95 9C 00 00
00098205 SW: 87 11 01 6A 4D 20 8D 91 80 40 EA C6 8E B3 C5 8E CE B1 51 99 02 90 00 8E 08 74
    72 02 00 57 C6 0A 7D 90 00
00000427 APDU: 0C B0 86 00 00 00 0E 97 02 00 00 8E 08 A4 50 5E 9F 14 17 7E 9F 00 00
00106568 SW: 87 11 01 D6 3C C8 A5 4C 68 18 4D 00 D0 CE A9 69 FD 29 6D 99 02 90 00 8E 08 4B
```

```
    96 77 3C C8 1F DF 56 90 00
00000527 APDU: 0C B0 87 00 00 00 0E 97 02 00 00 8E 08 D1 DE 23 D0 04 AE 97 12 00 00
00098482 SW: 87 11 01 DA 9D A2 62 60 23 28 93 AD 5D 8B 3E 80 F9 BF 15 99 02 90 00 8E 08 17
    69 4F A3 80 4E 60 9D 90 00
00000455 APDU: 0C B0 88 00 00 00 0E 97 02 00 00 8E 08 5B C4 9E 4F 9C D4 6A 7A 00 00
00108549 SW: 87 11 01 8E 99 29 C2 C4 CD E1 3A 67 E4 09 B0 36 D6 03 F0 99 02 90 00 8E 08 A2
    10 C6 64 96 04 2F 2A 90 00
00000431 APDU: 0C B0 89 00 00 00 0E 97 02 00 00 8E 08 1D C3 91 53 62 8F 99 7F 00 00
00099587 SW: 87 11 01 A0 8B AD 4C EE BA E0 53 F9 87 DA A9 62 91 D3 45 99 02 90 00 8E 08 08
    9B D8 10 6B A8 42 54 90 00
00000708 APDU: 0C B0 8A 00 00 00 0E 97 02 00 00 8E 08 77 38 E6 07 0A 3F 83 F3 00 00
00018276 SW: 99 02 6A 82 8E 08 00 AB 71 7E 9C 1B B5 4A 6A 82
00000612 APDU: 0C B0 8D 00 00 00 0E 97 02 00 00 8E 08 BD 93 9E 2A 24 AE A6 79 00 00
00109397 SW: 87 11 01 F2 AD 80 88 D5 79 7E 37 9E F0 31 FF 6B FF 8D C4 99 02 90 00 8E 08 02
    FA 3B B1 BE 8A 43 A8 90 00
00000682 APDU: 0C B0 91 00 00 00 0E 97 02 00 00 8E 08 88 9D 3B 3C B6 0F B9 0B 00 00
00102335 SW: 87 31 01 2D A3 FE F3 54 4E D7 D7 31 0E 7D C0 75 E9 CC EC FE A9 8E DA F3 DB 4C
    CC ED D3 99 C8 93 27 D0 14 D3 80 D6 7F 3E 75 A7 13 3C CE 44 FC DE 7A C6 5B 99 02 90
    00 8E 08 48 FC 71 32 26 96 3C 5E 90 00
00682768 APDU: 00 A4 00 0C 02 3F 00
00067207 SW: 90 00
00000640 APDU: 00 A4 02 0C 02 00 03
00005325 SW: 6A 82
00002123 APDU: 00 A4 00 0C 02 3F 00
00007911 SW: 90 00
00004905 APDU: 00 A4 02 0C 02 2F 00
00009087 SW: 90 00
00000638 APDU: 00 B2 04 04 FF
00004330 SW: 6D 00
00000489 APDU: 00 A4 00 0C 02 3F 00
00009842 SW: 90 00
00000615 APDU: 00 A4 02 0C 02 2F 02
00006068 SW: 6A 82
00000470 APDU: 00 A4 00 0C 02 3F 00
00007516 SW: 90 00
00000357 APDU: 00 A4 02 0C 02 2F 00
00008655 SW: 90 00
00000393 APDU: 00 B2 03 04 FF
00004646 SW: 6D 00
00000387 APDU: 00 A4 04 0C 0F F0 45 73 74 45 49 44 20 76 65 72 20 31 2E 30
00011582 SW: 6A 82
00000380 APDU: 00 A4 04 0C 0F D2 33 00 00 00 45 73 74 45 49 44 20 76 33 35
00009616 SW: 6A 82
00000333 APDU: 00 A4 04 00 08 27 60 00 12 1F 00 00 01
00009664 SW: 6A 82
00000308 APDU: 00 A4 00 0C 02 3F 00
00010695 SW: 90 00
00000344 APDU: 00 A4 02 0C 02 2F 00
00009648 SW: 90 00
00000365 APDU: 00 B0 00 00 FF
00010652 SW: 61 32 4F 0F E8 28 BD 08 0F A0 00 00 01 67 45 53 49 47 4E 50 0F 43 49 41 20 7A
    75 20 44 46 2E 65 53 69 67 6E 51 00 73 0C 4F 0A A0 00 00 01 67 45 53 49 47 4E 61 09
    4F 07 A0 00 00 02 47 10 01 61 0B 4F 09 E8 07 04 00 7F 00 07 03 02 61 0C 4F 0A A0 00
    00 01 67 45 53 49 47 4E 62 82
```

Literatur

[1] Bundesministerium des Inneren.
Bericht der Bundesregierung zur Verzichtbarkeit der Anordnungen der Schriftform und des persönlichen Erscheinens im Verwaltungsrecht des Bundes.
URL: https://www.bmi.bund.de/SharedDocs/Downloads/DE/Nachrichten/Pressemitteilungen/2016/07/bericht-schriftformerfordernisse.pdf (aufgerufen am 14. 09. 2016).

[2] Steffen Fründt. „Nur fünf Prozent nutzen den E-Perso im Internet“.
In: *Die Welt* (2015).
URL: http://www.welt.de/wirtschaft/article142059932/Nur-fuenf-Prozent-nutzen-den-E-Perso-im-Internet.html
(aufgerufen am 14. 09. 2016).

[3] Vergabestelle für Berechtigungszertifikate. *Erteilte Berechtigungszertifikate.*
URL: http://download.gsb.bund.de/VfB/npavfb.pdf (aufgerufen am 14. 09. 2016).

[4] Bundesministerium des Innern.
Bei welcher Aktion dürfen welche Daten übertragen werden? 2016.
URL: http://www.personalausweisportal.de/DE/Buergerinnen-und-Buerger/Der-Personalausweis/Details/DatenChip/datenChip_node.html
(aufgerufen am 14. 09. 2016).

[5] *AusweisApp2.* URL: https://www.ausweisapp.bund.de/startseite/ (aufgerufen am 14. 09. 2016).

[6] *PersoApp.* URL: http://www.persoapp.de/ (aufgerufen am 14. 09. 2016).

[7] *Open eCard App.* URL: https://www.openecard.org/startseite/ (aufgerufen am 14. 09. 2016).

[8] *Ausweis-Auskunft.* URL: https://www.buergerserviceportal.de/bund/ausweisapp/bspx_selbstauskunft (aufgerufen am 14. 09. 2016).

[9] *BSI TR-03110 Advanced Security Mechanisms for Machine Readable Travel Documents.* TR. Version 2.20. 2012.
URL: https://www.bsi.bund.de/DE/Publikationen/TechnischeRichtlinien/tr03110/index_htm.html (aufgerufen am 14. 09. 2016).

[10] Jens Bender, Marc Fischlin und Dennis Kügler.
„Security analysis of the pace key-agreement protocol“.
In: *International Conference on Information Security.* Springer. 2009, S. 33–48.

[11] *ISO 7816 - Part 4: Interindustry Commands for Interchange.* ISO. 2006.

[12] *ISO 7816 - Part 3: Cards with contacts – Electrical interface and transmission protocols.* ISO. 2006.

[13] *SIM Access Profile - Interoperability Specification*. Version V11r00. 2008. URL: https://www.bluetooth.org/docman/handlers/DownloadDoc.ashx?doc_id=158740 (aufgerufen am 14. 09. 2016).

[14] *Specification of the Bluetooth System*. Specification. Version 4.2. 2014. URL: https://www.bluetooth.org/DocMan/handlers/DownloadDoc.ashx?doc_id=286439&_ga=1.92506207.336146264.1467107304 (aufgerufen am 14. 09. 2016).

[15] *Specification for Integrated Circuit(s) Cards Interface Devices*. Version Revision 1.1. USB Implementers Forum, Inc. URL: http://www.usb.org/developers/docs/devclass_docs/DWG_Smart-Card_CCID_Rev110.pdf (aufgerufen am 14. 09. 2016).

[16] *stunnel*. URL: https://www.stunnel.org/index.html (aufgerufen am 14. 09. 2016).

[17] *vsmartcard*. URL: https://frankmorgner.github.io/vsmartcard/index.html (aufgerufen am 14. 09. 2016).

[18] *PCSC-Lite*. URL: http://pcsclite.alioth.debian.org/ (aufgerufen am 14. 09. 2016).

[19] *BSI TR-03119 Requirements for Smart Card Readers Supporting eID and eSign Based on Extended Access Control*. TR. Version 1.3. 2013. URL: https://www.bsi.bund.de/DE/Publikationen/TechnischeRichtlinien/tr03119/index_htm.html (aufgerufen am 14. 09. 2016).

Selbständigkeitserklärung

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig verfasst und noch nicht für andere Prüfungen eingereicht habe. Sämtliche Quellen einschließlich Internetquellen, die unverändert oder abgewandelt wiedergegeben werden, insbesondere Quellen für Texte, Grafiken, Tabellen und Bilder, sind als solche kenntlich gemacht. Mir ist bekannt, dass bei Verstößen gegen diese Grundsätze ein Verfahren wegen Täuschungsversuchs bzw. Täuschung eingeleitet wird.

Berlin, den 16. September 2016 .....

Diese CD enthält:

- Die vorliegende Bachelorarbeit als PDF
- Den Quellcode der Android-App