MobilePACE

Password Authenticated Connection Establishment implementation on mobile devices

Bachelor-Thesis von Moritz Horsch

September 2009

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Fachbereich Informatik
Kryptographie und Computeralgebra

MobilePACE
Password Authenticated Connection Establishment implementation on mobile devices

vorgelegte Bachelor-Thesis von Moritz Horsch

Betreuung:
Prof. Dr. Johannes Buchmann
Dr. Detlef Hühnlein
Dr. Alexander Wiesmaier

Tag der Einreichung:

CASED

Center for Advanced Security Research Darmstadt

Erklärung zur Bachelor-Thesis

Hiermit versichere ich die vorliegende Bachelor-Thesis ohne Hilfe Dritter nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Darmstadt, den 30. September 2009

_____

(Moritz Horsch)

Danksagung

Ich bedanke mich herzlich bei Prof. Dr. Johannes Buchmann, Dr. Detlef Hühnlein und Dr. Alexander Wiesmaier für die Möglichkeit mich mit diesem interessanten und innovativen Thema befassen zu können sowie für die hervorragende Zusammenarbeit und die sehr gute Betreuung.
Ebenso danke ich meiner Familie und meinen Freunden für die hilfreichen Ratschläge und das Korrekturlesen.
Ein ganz besonderer Dank gilt der Firma FlexSecure GmbH, insbesondere Manuel Hartl, für die Unterstützung, das Know-how und die Ressourcen, die mir bei der Ausarbeitung und Entwicklung dieser Arbeit zur Verfügung gestellt wurden.

Zusammenfassung

Die Verwaltung unser Identitäten, in Form von Benutzernamen und Passwörtern, entwickelt sich im elektronischen Zeitalter zu einer Herausforderung. Durch die Vielzahl der Angebote, die uns heute im Internet geboten werden, wird es immer schwieriger den Schutz unserer Identitäten, gemäß der Grundregeln der IT-Sicherheit, aufrecht zu erhalten.
Ab November 2010 steht den Bürgerinnen und Bürger der Bundesrepublik Deutschland ein neuer und innovativer Personalausweis zur Verfügung. Funktionen wie der elektronische Identitätsnachweis, die qualifizierte elektronische Signatur und der Einsatz von biometrischen Daten eröffnen eine Vielzahl neuer Anwendungsszenarien. Auf diese Weise bieten sich Möglichkeiten, die über die klassische Funktion als innerdeutsches Ausweisdokument weit hinaus gehen. Insbesondere im E-Government und E-Business sind eine Vielzahl attraktiver Einsatzmöglichkeiten geplant. Um die Schutzziele der Vertraulichkeit, Integrität und Verfügbarkeit sicherzustellen, kommt Extended Access Control (EAC) zum Einsatz. Bestehend aus dem Password Authenticated Connection Establishment (PACE) für den Aufbau einer sicheren und integeren Kommunikation über die Luftschnittstelle, der Terminal-Authentifizierung zum ausschließlichen Zugriff durch berechtigte Lesegeräte und der Chip-Authentifizierung, um die Echtheit des elektronischen Personalausweises zu gewährleisten bzw. zu bestätigen, stellt Extended Access Control ein Sicherheitssystem bereit, das die persönlichen und biometrischen Daten des Inhabers schützt, aber auch dessen Authentifizierung zuverlässig ermöglicht. Um die Funktionen des elektronischen Personalausweises gegen Missbrauch zu schützen, verfügt der elektronische Personalausweis über ein Passwort-Konzept, bestehend aus PIN, PUK usw.. Diese Protokolle und Verfahren ermöglichen einen effektiven und sicheren Schutz für kontaktlose Chipkarten, um das Sicherheitsrisiko einer unbemerkten Kommunikation über die kontaktlose Schnittstelle des elektronischen Personalausweises und dem unberechtigten Datenzugriff auf die persönlichen Daten zu verhindern.
Durch den Wunsch nach einem uneingeschränkten Zugang zu Informationen, zu jeder Zeit und an jedem Ort, setzen wir neue Anforderungen an die Hardwarekomponenten, Software und Bandbreiten. Die Betriebssysteme auf mobilen Endgeräten nähern sich immer stärker der Funktionalität moderner Desktopsysteme an. Kommunikationstechnologien im Personal Area Network (PAN) wie IrDA, Bluetooth, im Local Area Network (LAN) Wireless-LAN und im Wireless Wide Area Network (WWAN) UMTS, HSDPA stellen eine schnelle und komfortable Option des Informationsaustausches dar. Die Near Field Communication (NFC) Technologie setzt dabei neue Maßstäbe in der Kommunikation zwischen mobilen Geräten und physikalischen Objekten. Als Schnittstelle zur Radio Frequency Identification (RFID) Technologie eröffnen sich neue Anwendungsszenarien, wie die Kommunikation mit dem elektronischen Personalausweis. NFC weist nur eine geringe Übertragungsreichweite auf, profitiert aber von einem einfachen Verbindungsaufbau und einer intuitiven Bedienung. NFC-fähige Handys bzw. Smartphones bieten eine kostengünstige, attraktive und flexible Schnittstelle zwischen Anwendungen und dem elektronischen Personalausweis.
Als Prototyp einer Anwendung wurde das PACE Protokoll implementiert und die NFC Technologie als Kommunikationsschnittstelle zwischen dem Handy und dem elektronischen Personalausweis verwendet. Bei der Entwicklung stand insbesondere ein robustes, flexibles und leicht zu erweiterndes Design im Vordergrund.

Inhaltsverzeichnis

Abbildungsverzeichnis

„The best way to predict the future is to implement it.“
— *David Heinemeier Hansson*

Motivation

In einem Zeitalter, in dem der Schutz und der Nachweis der eigenen Identität zunehmend an Bedeutung gewinnt, scheinen elektronische Ausweisdokumente die Technologie der Zukunft und die Antwort auf die Probleme des Identitätsdiebstahls und der Realisierung eines vertrauenswürdigen und zuverlässigen Identitätsnachweises zu sein.
Eine Identifizierung von Personen geschieht zur Zeit durch Inaugenscheinnahme des Personalausweises und einem Abgleich der personenbezogenen Angaben und des Ausweisbildes. Im elektronischen Umfeld ist diese Art der Identifizierung aber nicht verwendbar. Eine Authentifizierung im Internet geschieht im Allgemeinen nur durch den Nachweis von Zugangsdaten, wie einem Benutzernamen und dem dazugehörigen Passwort. Für eine Authentifizierung gegenüber Applikationen im Internet wurden bereits Chipkarten entwickelt [1] [2]. Produkte und Zertifizierungsdienste sowie eine Rechtsgrundlage für die Verwendung und das Angebot elektronischer Signaturen sind in Deutschland bereits verfügbar. Demzufolge stehen die Technologien für die Herausforderungen, die elektronische Identitäten beinhalten, bereits zur Verfügung. Chipkarten für eine Authentifizierung oder zum Einsatz von elektronischen Signaturen setzten aber teilweise auf unterschiedliche Infrastrukturen und Techniken, was eine flächendeckende Nutzung verhindert. Unternehmen scheuen die hohen Investitionskosten und potenzielle Kunden sehen keine Verwendung, auch wenn die Vorteile offensichtlich sind.
Der elektronische Personalausweis (ePA oder ePerso) wird durch biometrische Daten des Inhabers nicht nur neue Maßstäbe bei hoheitlichen Ausweiskontrollen setzten, sondern auch durch die Funktionen des elektronischen Identitätsnachweises und der elektronischen Signatur eine breite Verwendung im privaten und geschäftlichen Umfeld bieten. Mit dem elektronischen Personalausweis steht die gewünschte homogene Infrastruktur und Technologie zur Verfügung, die Unternehmen die Sicherheit für Investitionen und Verbrauchern ein einheitliches System und ein breites Anwendungsspektrum bieten wird. Fakt ist, dass in 10 bis 15 Jahren die Mehrheit der Bürgerinnen und Bürger der Bundesrepublik Deutschland einen elektronischen Personalausweis besitzen werden, fraglich ist jedoch, ob die Bürgerinnen und Bürger die Kosten für neue Hardwarekomponenten, wie einem Lesegeräte für den Ausweis, tätigen werden. Ohne persönliche Anreize bzw. finanzielle Vorteile ist davon nicht auszugehen.
Eine andere Technologie könnte die Verbreitung der elektronischen Signatur und weiterer Funktionen des elektronischen Personalausweises beschleunigen. Near Field Communication (NFC) wird in mobilen Geräten verwendet und gestattet eine Kommunikation zwischen RFID-Chips bzw. Chipkarten, wie dem elektronischen Personalausweis, und einem mobilen Gerät. Eine starke Marktpräsenz von NFC-fähigen Mobilfunktelefonen ist zur Zeit zwar nicht gegeben, das Potenzial ist aber höher einzuschätzen. Als weitere Kommunikationstechnologie neben Bluetooth und Wireless-LAN setzt NFC auf eine etablierte Infrastruktur und die starke Präsenz mobiler Geräte. Ein NFC-fähiges Gerät dürfte demnach nicht nur kostengünstiger und attraktiver als ein stationäres Lesegerät sein, sondern auch eine flexiblere Nutzung gestatten. Mobil in Verbindung mit einem Notebook oder am heimischen Computer könnte ein NFC-fähiges Gerät die Schnittstelle zwischen Anwendung auf dem Computer bzw. im Internet und dem elektronischen Personalausweis darstellen.
Ein Problem beim Einsatz von kontaklosen Chipkarten wie dem elektronischen Personalausweis ist eine *unbemerkte* Kommunikation. Bei stationären Lesegeräten ist eine mögliche Kommunikation mit dem elektronischen Personalausweis im Allgemeinen noch offensichtlich, insbesondere weil eine externe Energieversorgung benötigt wird.
Ein NFC-fähiges Handy — als Lesegerät für kontaktlose Chipkarten — erweckt äußerlich jedoch den Anschein eines „normalen Handys“ und würde nicht als potenzielles Sicherheitsrisiko eingestuft. Das Sicherheitsrisiko ist demzufolge bei mobilen Lesegeräten deutlich höher einzuschätzen. Die Schutz-

ziele müssen demnach durch die Sicherheitsmechanismen des elektronischen Personalausweises realisiert werden und die Art des Lesegerätes, ob mobil oder stationär, darf keinen Einfluss auf diese haben. Als Sicherheitsmechanismus, zum Schutz der personenbezogenen und biometrischen Daten des Ausweisinhabers, kommt Extended Access Control (EAC) [3] zum Einsatz. Bestandteil von EAC und als Fundament des Sicherheitssystems des elektronischen Personalausweis ist das Password Authenticated Connection Establishment (PACE) Protokoll [3, Kapitel 4.2] einzuordnen. Ziel ist es, eine unbemerkte und unautorisierte Kommunikation zwischen dem elektronischen Personalausweis und einem Lesegerät auszuschließen, um die Luftschnittstelle als Hauptangriffspunkt abzusichern.
Im Rahmen dieser Arbeit stellen wir einen Prototyp einer Applikation für mobile Geräte vor, der in Verbindung mit einem NFC-fähigen Handy eine Kommunikation mit einem elektronischen Personalausweis aufbaut und den Ausweisinhaber authentifiziert.

1 Einführung

Die Technologien von Chipkarten, kontaktlosen Kommunikationsschnittstellen und biometrischen Ausweisdokumenten halten ab November 2010 Einzug in die Welt des Personaldokuments der Bürgerinnen und Bürger der Bundesrepublik Deutschland. Durch eine immer häufigere Nutzung des Internets zur Kommunikation, dem Erwerb und Verkauf von Waren und Dienstleistungen, sowie der Übermittlung hochsensibler Daten wie beim „Online-Banking“ wachsen die Anforderungen an die technischen Systeme. Auch die Bundesrepublik Deutschland setzt auf Innovation und versucht mit Projekten wie der elektronischen Steuererklärung[1] (ELSTER) oder der elektronischen Gesundheitskarte[2] (eGK) die neuen Technologien für Verwaltungsverfahren auf elektronischem Wege den Bürgerinnen und Bürger bereitzustellen.

> *„Jeder Mitgliedstaat* [der Europäischen Union] *sollte bis 2010 eine interoperable elektronische Identifizierung und Beglaubigung elektronischer Dokumente bereitstellen. Damit würden wir nicht nur das grenzüberschreitende E-Government befördern, sondern auch den Identitätsmissbrauch im Internet zurückdrängen.“ — Wolfgang Schäuble*

Dieses Zitat von Wolfgang Schäuble bei der Eröffnung der Konferenz *Advancing eGovernment*[3] am 1. März 2007 in Berlin verdeutlicht die Ziele des neuen elektronischen Personalausweises. Ein Problem beim Konsum der Angebote im Internet, die über das tägliche „Surfen“ und einer Kommunikation per E-Mail hinausgehen, ist der Nachweis der eigenen Identität. Der Beweis, wie beispielsweise der Inhaber eines Bankkontos zu sein, ist ein essenzieller Bestandteil jeglicher Nutzung solcher Angebote.
Durch die Vielzahl an Tätigkeiten, die wir heute über das Internet abwickeln können, steigt auch die Anzahl unserer partiellen Identitäten und deren Verwaltung. Eine partielle Identität ist eine Teilmenge der Eigenschaften unserer Identität in einem bestimmten Kontext. Eine E-Mail-Adresse ist eine partielle Identität im Bereich elektronischer Nachrichten und eine Kontonummer ist es im Kontext des Online-Banking-Angebots einer Bank. Bei vielen dieser partiellen Identitäten ist es wichtig, dass wir unsere eigene Identität nachweisen können, das heißt, dass wir der rechtmäßige Besitzer der partiellen Identitäten sind bzw. dass eine partielle Identität uns zugeordnet werden kann; in anderen Fällen wünschen wir eine solche Zuordnung nicht — wir sprechen dann von Anonymität. Da sich die von uns betrachteten partiellen Identitäten nur im elektronischen Kontext bewegen, lassen sich diese allgemein als elektronische Identitäten bezeichnen.
Heute existieren Banken, die ihren Service ausschließlich über das Internet anbieten und aus Kostengründen verschicken immer mehr Unternehmen Rechnungen und Vertragsunterlagen anstatt auf dem Postweg nur noch in elektronischer Form. Die Tendenz führt zu einer immer stärkeren Verlagerung der Verantwortung zum Kunden hin; der Service der Unternehmen beschränkt sich auf die Bereitstellung einer Kunden-Plattform im Internet. Für den Endverbraucher entsteht so eine Vielzahl von partiellen Identitäten, die er selbst verwalten und schützen muss. Zugangsdaten für die E-Mail-Adresse, für das Online-Banking, für den Stromanbieter, für den Telefon und Internet Provider usw. scheinen erst der Anfang dieser Entwicklung zu sein. Betrachtet man die Empfehlungen für „sichere“ Passwörter, d. h. Kombinationen aus zufälligen Buchstaben, Zahlen und Sonderzeichen sowie verschiedene Passwörter für unterschiedliche Konten, sind diese in Anbetracht der Fülle von partiellen Identitäten, die wir heute besitzen, nicht mehr umsetzbar. Die Folgen sind einfache oder notierte Passwörter, die mit den Grundregeln der IT-Sicherheit nicht vereinbar sind, weil Vertraulichkeit durch Verschlüsselung nur durch die

[1] http://www.elster.de
[2] http://www.die-gesundheitskarte.de
[3] http://www.eu2007.bmi.bund.de/nn_1035634/EU2007/DE/ServiceNavigation/Reden/content__Reden/BM_Schaeuble__Eroeffnung__Advanced__EGovernment.html

Geheimhaltung der verwendeten Passwörter Bestand hat. Ein weiterer Themenpunkt ist die Glaubwürdigkeit der elektronisch versendeten Dokumente. Ist die Rechnung von meinem Mobilfunkanbieter wirklich authentisch oder nur scheinbar identisch, aber mit einer abweichenden Bankverbindung versehen? Warum erhalte ich als Kunde Rechnungen nur per E-Mail, eine Kündigung meines Vertrags muss ich jedoch postalisch abwickeln? Solche Fragen stellen sich im Zuge der immer stärkeren Verbreitung der elektronischen Datenverarbeitung.
Die Bundesregierung möchte mit dem neuen elektronischen Personalausweis den Bürgerinnen und Bürgern die Verwaltung ihrer partiellen Identitäten erleichtern und auf diese Weise auch die IT-Sicherheit erhöhen. Das finanzielle Einsparpotenzial, das Unternehmen im E-Business mittels elektronischer Datenverarbeitung wie beispielsweise der „Online-Rechnung“ seit längerem umsetzten, plant auch die Bundesregierung für ihre staatlichen Behörden. Die unter dem Schlagwort E-Government bekannte Vision sieht eine digitale Verwaltung vor, die es den Bürgerinnen und Bürgern der Bundesrepublik Deutschland ermöglichen soll, eine Vielzahl verschiedener Tätigkeiten über das Internet abzuwickeln. Die Ummeldungen des Wohnortes, die Beantragung einer Kfz-Zulassung, die Anmeldung zum Führerschein usw. sind Anwendungsgebiete, die E-Government in Zukunft bieten wird. Ein Beispiel für bereits praktiziertes E-Government ist die elektronische Steuererklärung (ELSTER), über die jährlich fünf Millionen Einkommenssteuererklärungen bei den Finanzämtern eingereicht werden [4].
Der jetzige Personalausweis dient in erster Linie als Identitätsnachweis in hoheitlichen Kontrollen durch berechtigte staatliche Stellen. Mit dem Gesetzentwurf über Personalausweise und den elektronischen Identitätsnachweis [5] beschließt die Bundesregierung am 23.07.2008 [6] die klassische Aufgabe des Personalausweises als Identitätsnachweis um elektronische Funktionen zu erweitern. Die Merkmale des neuen elektronischen Personalausweises (ePA) sind der elektronische Identitätsnachweis, die qualifizierte elektronische Signatur und die Erweiterung der hoheitlichen Ausweisfunktion um biometrische Daten. Die neuen Funktionen eröffnen weitere Anwendungsgebiete und ermöglichen es, weitere Tätigkeiten vom heimischen Computer aus zu erledigen. Die technische Komponente des elektronischen Personalausweises stellt jedoch auch eine große Herausforderung an die IT-Sicherheit und die damit verbundenen Sicherheitsprotokolle und -mechanismen dar, denn das Ziel ist es, die IT-Sicherheit zu erhöhen und nicht durch die Erweiterung des elektronischen Personalausweises um technische Funktionen neue Sicherheitsrisiken zu schaffen.

Im nachfolgenden Kapitel gehen wir detailliert auf den elektronischen Personalausweis ein und betrachten die neuen Funktionen sowie die technischen Aspekte und Anwendungsmöglichkeiten. Wir erläutern die Sicherheitsmechanismen des bestehenden Personalausweises und stellen die technischen Sicherheitsverfahren vor, die bei elektronischen Ausweisdokumenten, wie dem ePA, zum Einsatz kommen. Neben den Sicherheitsprotokollen besitzt der elektronische Personalausweis ein Passwort-Konzept, das in Kapitel 2.2 erörtert wird. Die Grundlagen der verwendeten Sicherheitsprotokolle werden im dritten Kapitel vorgestellt und es wird auf das in diesem Dokument fokussierte Password Authenticated Connection Establishment (PACE) Protokoll detailliert eingegangen. In diesem Zusammenhang stellen wir das Password Authenticated Secure Channel (PASC) Protokoll vor und betrachten die Parallelen und Unterschiede zu PACE. Die Plattformen mobiler Endgeräte und Protokolle zur drahtlosen Kommunikation sind in Kapitel vier zusammengefasst. Das fünfte Kapitel befasst sich mit den technischen Grundlagen und Voraussetzungen für die Entwicklung und Implementierung des PACE Protokolls auf einem Handy. Welche neuen Szenarien und Perspektiven der elektronische Personalausweis und der entwickelte Prototyp eröffnet, wird abschließend in Kapitel sechs diskutiert.

2 Der elektronische Personalausweis

Das folgende Kapitel möchten wir den Neuerungen und Anwendungsszenarien des elektronischen Personalausweises widmen und insbesondere auf das Thema Sicherheit eingehen. Wir betrachten zuerst die neuen Funktionen des elektronischen Identitätsnachweises und der qualifizierten elektronischen Signatur (QES), geben dann auf das Thema Sicherheit ein und geben abschließend einen Ausblick auf die Anwendungsmöglichkeitengehen des elektronischen Personalausweises.
Die Verwendung des jetzigen Personalausweises als reines innerdeutsches Ausweisdokument hat sich bereits auf andere Gebiete ausgedehnt. In Teilen Europas fungiert er heute schon als Reisedokument und wird vielleicht in Zukunft auch diese Aufgabe für Reisen außerhalb der Staaten des Schengener Abkommens übernehmen.
Ähnlich dem elektronischen Reisepass (ePass) wird der elektronische Personalausweis (ePA) um biometrische Daten des Gesichts und — auf Wunsch des Inhabers — auch um Fingerabdrücke ergänzt. Sowohl im Bereich des E-Government als auch im E-Business soll die elektronische Übermittlung von Identitätsmerkmalen wie Name, Anschrift, Geburtsdatum usw. einen zuverlässigen Nachweis der Identität bieten. Möglich wäre anstelle des heutigen etablierten PIN und TAN-Verfahrens eine Authentifizierung beim Online-Banking mittels elektronischem Personalausweis. Die Übertragung von biometrischen Daten des Gesichts oder der optionalen Fingerabdrücke ist bei der elektronischen Kommunikation über das Internet nicht vorgesehen.
Eine qualifizierte elektronische Signatur (QES) [7] gemäß des Signaturgesetzes (SigG) [8] soll eine gleichwertige Lösung zur handschriftlichen Unterschrift bei rechtsverbindlichen Dokumenten in finanziellen und juristischen Bereichen bieten. Eine elektronische Signatur kann als Anhang eines elektronischen Dokuments gesehen werden, mit deren Hilfe die Identität des Unterzeichners nachgewiesen und Manipulationen erkannt werden können. Die Anforderungen, die eine qualifizierte elektronische Signatur laut SigG erfüllen muss, sind unter anderem die Identifizierung des Unterzeichners, keine nachträglichen Manipulationen und weitere Anforderungen an den Aussteller der Signatur. Da eine qualifizierte elektronische Signatur als Äquivalent zur handschriftlichen Unterschrift betrachtet wird und damit eine zweifelsfreie Zuordnung natürlicher Personen sichergestellt ist, hat diese auch vor Gericht Bestand. Das Unterzeichnen eines Dokuments mit einer qualifizierten elektronischen Signatur birgt demnach alle Rechte und Pflichten einer handschriftlichen Signatur in sich. Die Authentifizierung des Inhabers beim elektronischen Identitätsnachweis ist eine essenzielle Voraussetzung, die ebenfalls von der qualifizierten elektronischen Signatur erfüllt werden muss. Des Weiteren ist sicherzustellen, dass ein Dokument vor der elektronischen Unterzeichnung vollständig angezeigt wird und spätere Veränderungen (eines digital signierten Dokuments) erkannt werden. Der elektronische Personalausweis wird jedoch ohne qualifizierte elektronische Signatur ausgeliefert, was es dem Inhaber auf Wunsch ermöglicht, diese von einer Zertifizierungsstelle seiner Wahl zu beantragen. Dadurch wird vermieden, dass eine zentrale Stelle im privaten oder öffentlichen Bereich die Zertifizierung aller Bürgerinnen und Bürger vornimmt und verwaltet. Ferner existieren bereits Unternehmen mit einer eigenen Infrastruktur, die eine qualifizierte elektronische Signatur bereitstellen. Eine Übersicht[1] der Zertifizierungsdiensteanbieter (ZDA) veröffentlicht die Bundesnetzagentur.
Der elektronische Personalausweis (Abbildung 2.1) wird im Scheckkartenformat ID-1 nach ISO/IEC 7816 [9] gefertigt und reiht sich somit in das einheitliche Format der Kreditkarten und des neuen Führerscheins ein. In Bezug auf die Fälschungssicherheit sind von dem neuen ID1-Format keine Nach- bzw. Vorteile zu erwarten, die Bundesregierung sieht die Vorteile im handlichen Format, dass „das Mitführen des Personalausweises erleichtert und damit höhere Akzeptanz als das größere ID2-Format bei

[1] http://www.bundesnetzagentur.de/enid/Elektronische_Signatur/Zertifizierungsdiensteanbieter_ph.html

den Bürgerinnen und Bürgern genießt“ [4]. Äußerlich sind weiterhin persönliche Angaben wie Name, Geburtsdatum, Anschrift usw. ablesbar, diese werden jedoch auch auf dem integrierten Chip des elektronischen Personalausweises gespeichert. Diese Daten sind in der Machine Readable Zone (MRZ) abgelegt und durch ein Lesegerät erfassbar. Die Daten werden daher auch als MRZ-Daten bezeichnet. Die Abbildung des Gesichts wird im JPEG [10] oder JPEG2000 [11] Dateiformat gespeichert. Die Fingerabdrücke werden jeweils von dem linken und rechten Zeigefinger oder ggf. von dem Mittelfinger, Ringfinger oder Daumen genommen. Informationen über die Dateiformate und Auswirkungen bzw. Qualitätsverluste bei einer komprimierten Speicherung der Gesichtsbilder, sind den Artikeln [12] [13] zu entnehmen.

Abbildung 2.1.: Der elektronische Personalausweis
QUELLE: Bundesministerium des Innern - Deutschland

Der kontaktlose Radio Frequency Identification (RFID) Chip nach ISO/IEC 14443 [14] ist die wesentliche Innovation des elektronischen Personalausweises und im Gegensatz zur klassischen Magnetstreifenkarte in der Lage, die auf dem Chip befindlichen Daten über eine Entfernung von 10 bis 15 cm zu übertragen. Ein elektromagnetisches Feld, das von einem Lesegerät im 13,56 Mhz-Bereich erzeugt wird, dient zur Übertragung der Daten und zur Energieversorgung des elektronischen Personalausweises. Die Entscheidung für eine kontaktlose Chipkarte ist mit der längeren Haltbarkeit durch geringeren Verschleiß zu begründen. Kontaktlose Chipkarten sind darüber hinaus unempfindlich gegen Feuchtigkeit und Verschmutzung und in der Lage mit mobilen Endgeräten zu kommunizieren. Um die Funktionen und deren Verfügbarkeit des im elektronischen Personalausweis integrierten RFID-Chip innerhalb der Gültigkeitsdauer zu gewährleisten, scheidet die Technik einer klassischen Magnetstreifenkarte aus. Als Paradebeispiel einer Magnetstreifenkarte sei die EC- bzw. Kreditkarte genannt, die auf Grund der Abnutzungserscheinungen nur eine Gültigkeitsdauer von zwei Jahren besitzt. Durch die Wahl einer kontaktlosen Schnittstelle beim elektronischen Personalausweis kann die Gültigkeitsdauer von zehn Jahren beibehalten werden. Durch Herabsetzung der Gültigkeitsdauer werden erhebliche Mehrkosten und Verwaltungsarbeit für Bürgerinnen und Bürger sowie für die Behörden vermieden.

Kontaktlose Chipkarten erfordern jedoch neue Anstrengungen zum Schutz der auf der Karte gespeicherten Daten. Während die Verwendung einer kontaktbehafteten Chipkarte (z.B. klassische Magnetstreifenkarte) den Besitz der Karte erfordert, ist diese Voraussetzung bei kontaktlosen Karten nicht gegeben. Hierbei ist es ausreichend, wenn sich die kontaktlose Karte innerhalb des Übertragungsradius befindet, wodurch eine unbemerkte und ein Abhören der Kommunikation möglich ist. Bei kontaktbehafteten Chipkarten ist eine Authentifizierung des rechtmäßigen Besitzers nötig, diese Anforderung ist bei kontaktlosen Chipkarten essenziell, um eine missbräuchliche Nutzung auszuschließen. Um das Sicherheitsniveau auch bei kontaktlosen Karten wie dem elektronischen Personalausweis aufrecht zu erhalten, werden die in Kapitel 2.1 und 2.2 vorgestellten Protokolle und Verfahren eingesetzt.

Die Investitionskosten des elektronischen Personalausweises beziffert die Bundesregierung auf 7,75 Millionen Euro [5], bestehend aus Softwareentwicklung 1,71 Mio., Infrastruktur 4,02 Mio. und Hardwarekomponenten 2,02 Mio. Euro. Für die Wirtschaft erwartet sie eine Nettoentlastung von 123,29 Mio. Euro [15]. Die Kosten für Bürgerinnen und Bürger sind noch offen, werden sich aber vermutlich an denen des elektronischen Reisepasses von zur Zeit 59 Euro orientieren. Für die Nutzung des elektronischen Identitätsnachweises und der qualifizierten elektronischen Signatur wird ein Lesegerät benötigt, dessen Preispanne zum jetzigen Zeitpunkt zwischen 50 und 100 Euro liegt. Durch den großen Absatzmarkt, nach Einfügung des elektronischen Personalausweis ist zu erwarten, dass die Lesegeräte günstiger werden.

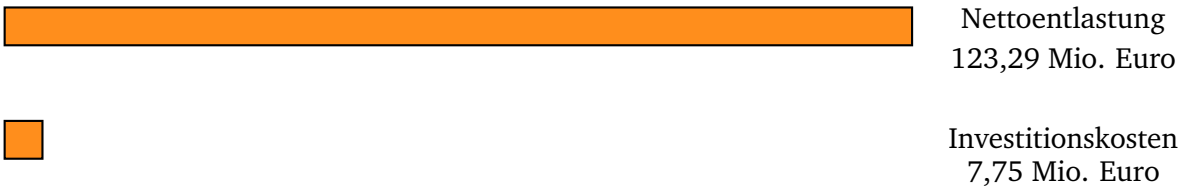

Abbildung 2.2.: Kosten und Einsparpotenzial

2.1 Sicherheitsmerkmale und Schutzmechanismen

Der jetzige Personalausweis zeichnet sich vor allem durch optische Sicherheitsmerkmale aus, um eine Fälschung durch Kopieren oder Manipulationen zu verhindern. Die Merkmale reichen von Hologrammen und einer Oberflächenprägung des Ausweisdokuments, bis zu einer speziellen Drucktechnik. Diese Maßnahmen zur Verbesserung der Dokumentensicherheit sind in der Verordnung (EG) Nr. 2252/2004 [16] definiert. Eine genaue Auflistung der Sicherheitsmerkmale für deutsche Ausweisdokumente ist im Anhang in Abbildung A.1 verfügbar. Die Sicherheit, Glaubwürdigkeit und Echtheit des jetzigen Personalausweises beruht auf diesen optischen Sicherheitsmerkmalen, beim elektronischen Personalausweis werden diese Eigenschaften durch technische Methoden realisiert bzw. erweitert. Die elektronische Speicherung digitaler Daten auf dem elektronischen Personalausweis bedarf daher elektronischer Sicherheitsmechanismen zur Gewährleistung der Schutzziele. Das wichtigste Ziel, ist der Erhalt des bestehenden hohen Sicherheitsstandards. Die Anforderung ist dabei, „nur mit kaum vertretbarem logistischen, technischen und finanziellen Aufwand“ [4], den elektronischen Personalausweis zu manipulieren oder zu kopieren. Der Personalausweis hat in den letzten Jahren durch den Schengen-Raum und die Nutzung des Ausweises als Reisedokument an Bedeutung gewonnen. Um den Identitätsmissbrauch zu verhindern, ist das Ziel, durch biometrische Merkmale den Ausweis stärker an dessen Inhaber zu binden. Durch ein immer stärker zusammenwachsendes Europa soll der elektronische Personalausweis nicht nur die Aufgabe eines Identitätsnachweises in Deutschland bieten, sondern in allen EU-Mitgliedsstaaten.

„Computer security rests on confidentiality, integrity, and availability“ — Matt Bishop [17]

Die drei Hauptprinzipien der IT-Sicherheit Vertraulichkeit, Integrität und Verfügbarkeit spielen beim elektronischen Personalausweis eine wichtige Rolle. Das Schutzziel der Authentizität kann als Teil der Integrität betrachtet werden [17, Kapitel 1.1.2]. Zur Umsetzung dieser Schutzziele wurden eine Reihe von Sicherheitsprotokollen zur sicheren Kommunikation zwischen dem elektronischen Personalausweis und einem Lesegerät sowie zur Zugriffskontrolle auf die Daten entwickelt. Die auf dem ePA gespeicherten Daten lassen sich in die Kategorien *weniger sensibel* und *sensibel* einordnen, wobei die MRZ-Daten allgemein als weniger sensibel eingestuft werden, weil diese ohnehin auf dem ePA aufgedruckt sind. Die gespeicherten biometrischen Daten, das Ausweisbild und die optionalen Fingerabdrücke sind sensible Daten und bedürfen deshalb besonderer Schutzmaßnahmen. Die International Civil Aviation Organization (ICAO) definiert mehrere Sicherheitsprotkolle [18], um die Schutzziele für biometrische bzw. elektronische Ausweisdokumente wie dem elektronischen Reisepass und dem elektronischen Personalausweis zu realisieren:

- **Passive Authentifizierung**
  Die Authentizität und Integrität steht bei der passiven Authentifizierung im Vordergrund. Um die Integrität der Daten auf dem elektronischen Personalausweis zu überprüfen, berechnet das Lesegerät eine Prüfsumme (Hash-Wert) der Datengruppen. Dieser Hash-Wert wird dann mit dem auf dem elektronischen Personalausweis gespeicherten Document Security Object (SO_D) verglichen. Die Prüfsumme wurde vom Aussteller (Document Signer) des elektronischen Personalausweises berechnet, mit dessen privatem Schlüssel signiert und im Document Security Object gespeichert. Die Integritätsprüfung schließt jedoch nicht aus, dass einzelne oder mehrere Datengruppen und das Document Security Object manipuliert wurden. Es ist also erforderlich, die Authentizität, d. h. die Echtheit der Daten, zu überprüfen. Das Lesegerät extrahiert aus dem Document Security Object den Document Signer, der bei der Erstellung des Ausweises das Document Security Object signiert hat. Mit Hilfe des Zertifikats des Document Signers überprüft das Lesegerät die Signatur des Document Security Object. Unter der Annahme, dass die Signatur nicht gefälscht werden kann, könnte ein manipulierter elektronischer Personalausweis einer Integritätsprüfung standhalten, jedoch keiner Kontrolle der Echtheit der Daten. Durch die passive Authentifizierung ist das Lesegerät in der Lage, manipulierte Ausweisdokumente zu erkennen. Würden einzelne oder mehrere Attribute wie beispielsweise Name oder Geburtsdatum der MRZ-Daten unautorisiert verändert werden,

stimmt die vom Lesegerät berechnete Prüfsumme nicht mehr mit dem Document Security Object überein und die Signatur kann nicht verifiziert werden. Die passive Authentifizierung ist jedoch nicht in der Lage, vollständig kopierte bzw. geklonte Ausweise zu erkennen. Diese Aussage muss jedoch, wie in [3] angemerkt wird, differenzierter betrachtet werden. Das Klonen eines Ausweises impliziert nicht, dass dieser auch physikalisch kopiert wurde. Es ist also denkbar, dass ein Ausweis optisch, d. h. aufgedruckt, andere persönliche Daten aufweist, als auf dem Chip gespeichert sind. Ein Lesegerät wäre mittels passiver Authentifizierungen in der Lage einen geklonten elektronischen Personalausweis zu erkennen, wenn es die optischen MRZ-Daten mit den in Datengruppe 1 digital gespeicherten Daten vergleicht. Um die Sicherheit des Ausweises zu garantieren, müsste dann allerdings physikalisch-optische Manipulationen ausgeschlossen werden.

- **Aktive Authentifizierung**
  Die passive Authentifizierung ermöglicht nur eine Überprüfung der Integrität und Authentizität der Daten, jedoch keine Überprüfung des Chips. Die Kontrolle der Einzigartigkeit des Chips ist die Aufgabe der aktiven Authentifizierung, um auch geklonte Ausweise zu identifizieren. Der elektronische Personalausweis muss dem Lesegerät nachweisen, dass er einen individuellen und einmaligen privaten Schlüssel besitzt. Dieser Schlüssel ist im geschützten und nicht zugänglichen Bereich des Chips gespeichert, das heißt, er kann nicht ausgelesen und demnach nicht kopiert werden. Bei dem Klonen eines Ausweises könnten zwar die Informationen der einzelnen Datengruppen kopiert werden, jedoch nicht der private Schlüssel des Chips. Bei der aktiven Authentifizierung sendet das Lesegerät eine Zufallszahl an den Chip, der diese mit seinem privaten Schlüssel signiert. Das Lesegerät verifiziert die Signatur anhand des in Datengruppe 15 gespeicherten öffentlichen Schlüssels. Die aktive Authentifizierung bestätigt die Integrität und Authentizität des Chips und des Document Security Object, bietet jedoch keine vertrauliche und integere Kommunikation zwischen Lesegerät und Chip.

- **Chip-Authentifizierung**
  Die Chip-Authentifizierung ist der aktiven Authentifizierung ähnlich, weist jedoch zwei Vorteile auf: Das Erstellen von Bewegungsprofilen ist nicht möglich, weil der Chip nicht gutgläubig Zufallszahlen signiert. Des Weiteren werden Sitzungsschlüssel für das Signieren und Verschlüsseln berechnet, die in der weiteren Kommunikation für eine vertrauliche und integere Verbindung sorgen, dem sogenannten *Secure Messaging*. Die Ziele sind identisch zur aktiven Authentifizierung, das Überprüfen, ob der Chip geklont bzw. ausgetauscht wurde. Dabei kommt das klassische oder das auf elliptischen Kurven basierende Diffie-Hellman Protokoll zum Einsatz (siehe Kapitel 3). Der Chip und das Lesegerät tauschen Informationen für die Berechnung eines gemeinsamen Schlüssels aus, der dann zum Ableiten von Sitzungsschlüsselen verwendet wird. Der Protokollablauf der Chip-Authentifizierung ist im Anhang in Abbildung A.2 illustriert. Nach erfolgreicher Chip-Authentifizierung muss eine passive Authentifizierung durchgeführt werden, das heißt, dass das Lesegerät überprüfen muss, ob der öffentliche Schlüssel authentisch ist.

- **Terminal-Authentifizierung**
  Um Zugang auf die sensiblen Daten, wie Fingerabdrücke oder das Bild des Inhabers des elektronischen Personalausweises, zu erhalten, muss das Terminal nachweisen, dass es zu diesem Datenzugriff berechtigt ist. Die Zugriffskontrolle erfolgt mit Zertifikaten, die von den jeweiligen staatlichen Behörden ausgestellt werden. Ist ein Lesegerät in der Lage, ein Zertifikat mit den nötigen Berechtigungen gegenüber dem Chip nachzuweisen und ist es dem Chip möglich, dieses zu prüfen, erhält das Lesegerät Zugriff auf die sensiblen Daten. Der Protokollablauf ist im Anhang in Abbildung A.3 verdeutlicht. Nach erfolgreicher Terminal-Authentifizierung muss der Chip die gesicherte Verbindung mit dem öffentlichen Schlüssel des Terminals verknüpfen, damit eine Kopplung der Zugriffsrechte an das Lesegerät bestehen bleibt.

Die passive und aktive Authentifizierung dienen jedoch ausschließlich der Überprüfung auf Integrität und Authentizität des Ausweises und des Lesegeräts, sie bieten jedoch keinen Schutz vor unautorisiertem Zugriff auf die Daten. In diesem Zusammenhang wurde das Sicherheitssystem Extended Access Control (EAC) [3] entwickelt. In der ersten Version kam EAC bei dem elektronischen Reisepass zum Einsatz und sollte die Ziele einer sicheren Kommunikation und Zugriffskontrolle auf die digital gespeicherten Daten erfüllen. EAC in der Version 1 (Abbildung 2.3) besteht aus Basic Access Control (BAC) [3, Anhang H], Terminal-Authentifizierung (TA) [3, Kapitel 4.4] und Chip-Authentifizierung (CA) [3, Kapitel 4.3]. BAC dient dem Aufbau einer sicheren Kommunikation zwischen Lesegerät und dem ePass sowie der Kompensation des Nachteils einer kontaktlosen Chipkarte, d. h. der Überprüfung ob, ein Lesegerät optischen Zugang zum Ausweis hat und keine unbemerkte Kommunikation stattfindet. Extended Access Control in der Version 1 brachte jedoch einige Sicherheitsprobleme mit sich. Die Sicherheit von BAC kann durch Kryptoanalyse erheblich reduziert werden und die Abfolge von Chip-Authentifizierung und Terminal-Authentifizierung ermöglicht das Erstellen von Bewegungsprofilen des Passinhabers. Das BAC Protokoll benutzt zur Berechnung des Schlüssels für die sichere Kommunikation die MRZ-Daten. Durch einfache Kryptoanalyse lässt sich jedoch die Entropie, d. h. die Anzahl aller möglichen Schlüssel, erheblich reduzieren, wie im Artikel *Crossing Borders: Security and Privacy Issues of the European e-Passport* [19] anschaulich gezeigt wird. Für deutsche Ausweisdokumente lässt sich so die Anzahl von möglichen Schlüsseln von ca. 2^{56} auf 2^{35} verringern wie die Firma Riscure [20] illustriert hat. Verdeutlicht wird das Ausmaß durch die Tatsache, dass die National Institute of Standards and Technology (NIST) eine Schlüssellänge von 80 Bits, d. h. eine Entropie von 2^{80}, empfiehlt [21] und bei heutigem Online-Banking 128-Bit als Standard gelten.

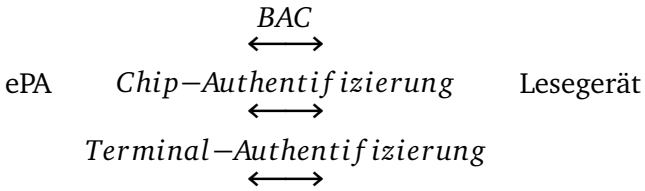

Abbildung 2.3.: Extended Access Control (EAC) Version 1

Aufgrund dieser Schwächen wurde Extended Access Control für den elektronischen Personalausweis überarbeitet und kommt in der Version 2.01 (Abbildung 2.4) zum Einsatz. Der wesentliche Schritt zur Erhöhung der Sicherheit ist der Einsatz des Password Authenticated Connection Establishment (PACE) Protokolls im Gegensatz zu Basic Access Control (BAC). Die Terminal-Authentifizierung und Chip-Authentifizierung wurde nur in Details überarbeitet, allerdings hat sich die Reihenfolge der Protokolle geändert. In der Version 1 von EAC wurde die Chip-Authentifizierung vor der Terminal-Authentifizierung ausgeführt, wohingegen sich in EAC Version 2.01 zuerst das Terminal bzw. Lesegerät gegenüber dem elektronischen Personalausweis authentifizieren, erst dann erfolgt die Chip-Authentifizierung. Die Terminal-Authentifizierung wird also vor der Chip-Authentifizierung ausgeführt, um so die Möglichkeit der Erstellung von Bewegungsprofilen zu verhindern.

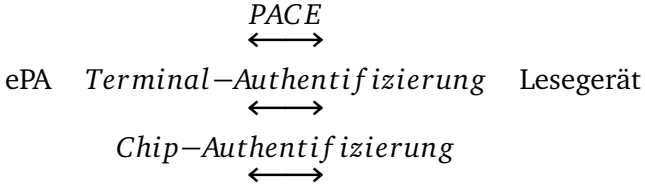

Abbildung 2.4.: Extended Access Control (EAC) Version 2.01

2.2 Passwort-Konzept

Bei den Funktionen des elektronischen Identitätsausweises und der qualifizierten elektronischen Signatur sind Sicherheitsmechanismen auf Grundlage des Besitzes nicht ausreichend. Analog zu EC- und Kreditkarten bedarf es eines geheimen Passwortes, das nur dem Inhaber bekannt ist und anhand dessen er den rechtmäßigen Besitz der Karte nachweisen kann. Bei den einzelnen Passwörtern des elektronischen Personalausweises wird zwischen *blockierend* und *nichtblockierend* unterschieden. Bei blockierenden Passworten wird nach mehrmaliger Falscheingabe der Zugriff gesperrt. Nichtblockierende Passwörter können hingegen beliebig oft falsch eingegeben werden. Die einzelnen Passwörter für eine Authentifizierung gegenüber dem elektronischen Personalausweis möchten wir im folgenden Abschnitt vorstellen:

- **Personal Identification Number (PIN)**
  Die PIN ist ein blockierendes Passwort und sollte nur dem Inhaber bekannt sein. Durch die Eingabe der PIN authentisiert sich der Inhaber gegenüber Lesegeräten oder Online-Applikationen, d. h. er weist den rechtmäßigen Besitz der Karte nach. Vorgesehen ist eine sechsstellige PIN.

- **Personal Unblocking Key (PUK)**
  Durch mehrmalige Falscheingabe der PIN kann der ePA eine Authentifizierung mittels PIN verweigern, die PIN ist dann gesperrt. Der PUK kann dann dazu verwendet werden, die PIN wieder zu entsperren. Der PUK ist ein nichtblockierendes Passwort und sollte nur dem Inhaber bekannt sein und vertraulich behalten werden.

- **Card Access Number (CAN)**
  Diese Kartenzugriffsnummer ist auf dem elektronischen Personalausweis aufgedruckt und maschinenlesbar, d. h. ein Lesegerät ist in der Lage, die CAN vom Ausweis abzulesen. Durch die CAN kann ein Lesegerät bestätigen, dass es optischen Zugriff auf den Ausweis hat. Die Verwendung von PACE mit der CAN impliziert jedoch ein erhebliches Sicherheitsrisiko: Ist einem Dritten die CAN bekannt, kann diese als Grundlage für einen Man-in-the-Middle-Angriff missbraucht werden.

Abbildung 2.5.: Card Access Number
(Fotomontage)
QUELLE: Bundesministerium des Innern - Deutschland

- **Machine Readable Zone (MRZ)**
  Das MRZ-Passwort ist ein nichtblockierendes Passwort. Auf dem RFID-Chip ist es in einem geschützten Bereich gespeichert und vom Lesegerät wird das MRZ-Passwort aus der Dokumentennummer, dem Geburtsdatum und dem Ablaufdatum des Ausweises berechnet. Ein Passwort auf Basis der MRZ-Daten kann sowohl bei PACE als auch bei BAC verwendet werden.

Um das Erraten der PIN durch Ausprobieren aller möglichen Kombinationen zu erschweren, verfügt der elektronische Personalausweis über einen Fehlbedienungszähler (FBZ). Die einzelnen Zustände des Passwort-Konzepts sind in Abbildung 2.6 illustriert. Der Versuch die korrekte PIN oder das Passwort durch schlichtes Ausprobieren aller Möglichkeiten zu erlangen, wird als Brute-Force Angriff [22] bezeichnet.
Bei einer sechsstelligen PIN würde es sich um 10^6 mögliche Kombinationen handeln. Durch diesen Zähler werden verschiedene Zustände des Passwort-Konzepts modelliert. Generell ist die Nutzung der PIN möglich, sie wird jedoch nach mehrmaliger Falscheingabe blockiert. Erst durch die auf dem elektronischen Personalausweis aufgedruckte CAN und der danach folgenden korrekten Eingabe der PIN kann der Ausweis entsperrt werden. Die Kopplung an die CAN dient folgendem Sachverhalt: Durch die Eigenschaften einer kontaktlosen Chipkarte des ePA wäre es auf kurze Distanz möglich — vom Inhaber unbemerkt — die PIN mehrmals falsch einzugeben und den Ausweis so zu sperren. Bei einem blockierten elektronischen Personalausweis wird eine Falscheingabe der CAN ignoriert. Wird jedoch nach korrekter CAN die falsche PIN eingegeben ist ein PUK zum entsperren nötig.

Durch den Fehlbedienungszähler (FBZ) wird ein verzögertes Blockieren modelliert, um einen Brute-Force Angriff zu erschweren. Wie in Abbildung 2.6 illustriert kann sich der FBZ in drei Zuständen befinden: FBZ > 1, FBZ $= 1$ und FBZ $= 0$. Im ersten Zustand kann die PIN uneingeschränkt verwendet werden. Im Zustand FBZ $= 1$ ist die PIN außer Kraft gesetzt und muss in der gleichen Sitzung durch die korrekte Eingabe der CAN freigeschaltet werden. Im dritten Zustand ist die PIN gesperrt und muss durch die korrekte Eingabe der PUK freigeschaltet werden. Durch die korrekte Eingabe der PIN im ersten und zweiten Zustand bzw. der PUK im dritten Zustand wird der Fehlbedienungszähler zurückgesetzt.

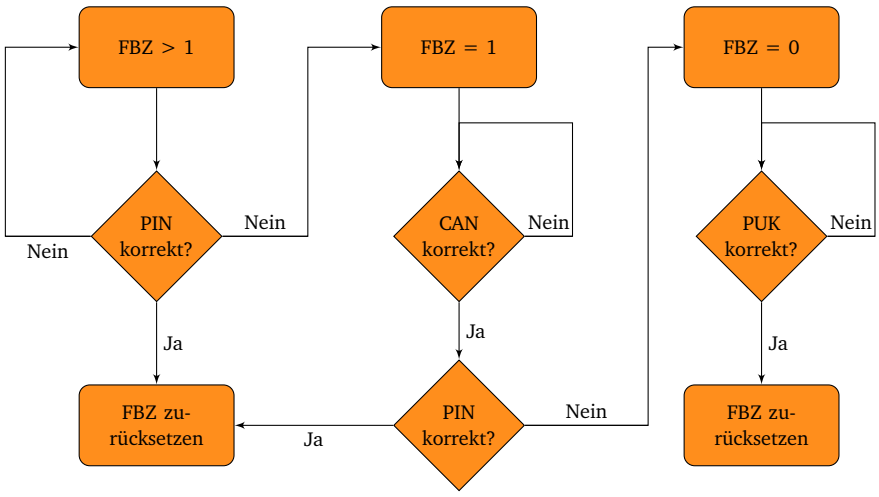

Abbildung 2.6.: Fehlbedienungszähler (FBZ) [23]

2.3 Anwendungsgebiete

In der Einführung wurden bereits einige neue Anwendungsszenarien angesprochen. Um das Potenzial des elektronischen Personalausweises zu verdeutlichen, gehen wir auf diese noch etwas genauer ein.

> *„Mit dem elektronischen Personalausweis steht eine neue IT-Infrastruktur zur Verfügung. Anwendungen im eGovernment, eBusiness und eCommerce werden einfacher nutzbar sein. Unser Bestreben muss es sein, schon mit der Einführung des neuen Ausweises eine Vielzahl attraktiver Einsatzmöglichkeiten für die Bürgerinnen und Bürger anbieten zu können. Deshalb meine Bitte an die IT-Dienstleister, Auktionsplattformen, Online-Häuser, Banken und andere Interessenten: Lassen Sie gemeinsam eine breite Palette von Anwendungen entwickeln und testen, den Nutzen für alle Beteiligten transparent machen und so in unser aller Interesse Akzeptanz schon im Vorfeld der Einführung schaffen.“ — Dr. Hans Bernhard Beus (Staatssekretär im Bundesministerium des Innern)*

2.3.1 E-Government

Die gesamte Verwaltung staatlicher Behörden geschieht heute in elektronischer Form. Formulare, Anträge und An- und Abmeldungen werden aber zum größten Teil noch in Papierform eingereicht, dabei ist der Zeitaufwand immens, diese in elektronische Systeme einzugeben, und Fehler bei der Eingabe bzw. Erfassung der Formulare sind nicht auszuschließen.
Die Formulare erhalten Bürgerinnen und Bürger zwar schon in digitaler Form auf den Internetplattformen der jeweiligen Behörden, aber ein Ausdruck und eine handschriftliche Unterzeichnung ist unumgänglich. Für die elektronische Einreichung der Formulare fehlt die digitale Unterschrift. Für die elektronische Steuererklärung (ELSTER) ist die Beantragung eines Zertifikats notwendig, mit dem die Einkommenssteuererklärung in elektronischer Form eingereicht werden kann. Würde dieses Verfahren für weitere Projekte angewandt, hätten wir nicht nur eine Vielzahl von partiellen Identitäten im privaten Bereich, sondern zusätzlich noch im Kontext des E-Government.
Das Ziel ist es, ein einheitliches System bzw. Verfahren zum Identitätsnachweis und zur digitalen Unterschrift zu etablieren. Diese Aufgabe wird in Zukunft der elektronische Personalausweis mit dem elektronischen Identitätsnachweis und der qualifizierten elektronischen Signatur bieten. Künftig sollen beispielsweise Bürgerinnen und Bürger bei einem Umzug ihrer Meldepflicht über das Internet nachkommen können. Der elektronische Personalausweis dient als Identitätsnachweis und mittels der qualifizierten elektronischen Signatur wird das Formular unterschrieben und kann durch die staatlichen Behörden auf Integrität und Authentizität geprüft werden. Ein weiteres Szenario ist der Zugriff auf persönliche Daten wie der eigene Schufa-Eintrag oder das Punktekonto im Verkehrszentralregister. Mittels des elektronischen Identitätsnachweises könnten diese Informationen den Bürgerinnen und Bürgern auf einer Internetplattformen zugänglich gemacht werden.
Zusammenfassend lässt sich sagen, dass der elektronische Personalausweis im Bereich E-Government die Kosten für den Ausdruck und Versand der Formulare und die Wartezeiten bei den Behörden reduzieren wird. Behördengänge würden entfallen und Bürgerinnen und Bürger können ihre Anliegen unabhängig von den Öffnungszeiten erledigen. Im Gegensatz fallen jedoch Kosten für den Erwerb der benötigten Hardware und eine Einarbeitungszeit in die jeweilige Software an. Für die staatlichen Behörden sieht die Bundesregierung eine schnelle und fehlerfreie elektronische Datenverarbeitung und infolgedessen eine schnellere Bearbeitung der Anliegen der Bürgerinnen und Bürger.

2.3.2 E-Business

Das Thema des „Online-Banking“ wurde bereits angesprochen. Neben der reinen Authentifizierung mit Hilfe des elektronischen Personalausweis, könnte die Technik des elektronischen Personalausweises eine Konto-Eröffnung im Internet ermöglichen [24] [25]. Ein Vorzeigen des Personalausweises bei der Bank, aber auch das Postident-Verfahren[2] würden sich erübrigen. Eine Konto-Eröffnung könnte demnach schneller und sicherer erfolgen, weil die persönlichen Daten verschlüsselt über das Internet übertragen werden können und nicht auf dem Postweg. Der elektronische Personalausweis ermöglicht in diesem Zusammenhang auch die Identifizierung nach dem Geldwäschegesetz (§ 4 GwG) [26]. Das wesentliche Sicherheitsrisiko des heutigen PIN und TAN Verfahrens ist das Ausspähen der Kontodaten durch Trojaner auf dem eigenen Computer oder durch Manipulation der Geldautomaten. Des Weiteren werden ganze Internetauftritte von Banken kopiert und Kunden dazu gebracht ihre Kontodaten auf einer gefälschten Internetseite einzugeben, die dann missbraucht werden. Diese sogenannten Phishing-Attacken könnten durch den Einsatz des elektronischen Personalausweises verhindert werden [4].
Mit der Verwendung eines Lesegerätes mit Keypad (Ziffernblock) wäre die PIN-Eingabe unabhängig vom eigenen Computer und durch die Überprüfung des Zertifikats der Internetplattform wären solche Angriffe nicht mehr möglich. Der elektronische Identitätsnachweis des ePA ist aber nicht nur auf den Einsatz im Internet beschränkt. Zugangskontrollen zu Gebäuden oder sicherten Bereichen könnten ebenfalls mit dem elektronischen Personalausweis realisiert werden. Denkbar wäre auch die Verwendung als Autoschlüssel.
Die Stärke des elektronischen Identitätsnachweis liegt vor allem in der direkten Erfassung der persönlichen Daten wie Name, Anschrift usw. durch elektronische Systeme. Das Problem des Medienbruchs, d. h. der Wechsel des informationstragenden Mediums, wie beispielsweise durch „abtippen“ der Daten kann damit behoben werden. Seit dem 1. Januar 2007 existiert das Gesetz über das elektronische Handelsregister, Genossenschaftsregister, sowie das Unternehmensregister (EHUG), in den veröffentlichungspflichtige Unternehmensdaten zentral publiziert werden. Des Weiteren sind alle Kapitalgesellschaften dazu verpflichtet ihre Jahresabschlüsse im Bundesanzeiger[3] zu veröffentlichen. Das Veröffentlichen dieser Informationen soll ab 2010 ausschließlich mittels qualifizierter elektronischer Signatur erfolgen, um die Integrität und Authentizität der Daten zu gewährleisten. Ein weiterer Anwendungsfall für die qualifizierte elektronische Signatur, der schon heute Verwendung findet, ist der Versand von Rechnungen per E-Mail. Insbesondere Unternehmen setzen diese Technologie zum Signieren der versendeten „Online-Rechnungen“ ein, um ihren Kunden beispielsweise das elektronische Einreichen zum Vorsteuerabzug beim Finanzamt zur ermöglichen. Mit Hilfe des elektronischen Personalausweises lassen sich diese Dokumente digital signieren und der Empfänger ist in der Lage die Authentizität der Rechnung zu prüfen.

[2] http://www.deutschepost.de/dpag?xmlFile=1015469
[3] http://www.ebundesanzeiger.de

2.3.3 Zusammenfassung

Abschließend stellen wir fest, dass der elektronische Identitätsnachweis eine fehlerfreie und schnelle Erfassung der persönlichen Daten, sowie eine effiziente Identitätsprüfung des Ausweisinhabers ermöglicht. Das autorisierte Auslesen dieser Daten wird mittels Berechtigungszertifikaten überprüft und eingeschränkt und für den Inhaber besteht durch das Passwort-Konzept eine effiziente Möglichkeit zur Einwilligung in diesem Vorgang. Die qualifizierte elektronische Signatur bietet nicht nur die Möglichkeit einer digitalen Unterschrift, sondern auch das Protokollieren von Zugriffen auf sensible Daten wie Geschäftsgeheimnisse oder, wie im Beispiel des ELENA-Verfahrens [27], auf vertrauliche Informationen von Arbeitnehmern.

Der Vollständigkeit wegen möchten wir noch das Thema E-Commence ansprechen. Hier liegt die Stärke des elektronischen Personalausweises insbesondere im elektronischen Identitätsausweis. Eine Authentifizierung beim Online-Banking lässt sich ebenfalls auf Online-Shops, Webmail usw. übertragen. Im Gegensatz zur Authentifizierung mittels Benutzername und Passwort könnte diese sicherer durch den elektronischen Identitätsausweis erfolgen.

Folgende Abbildung 2.7 illustriert mögliche Anwendungsszenarien des elektronischen Personalausweises. Details sind in dem Dokument *Einführung des elektronischen Personalausweises in Deutschland - Grobkonzept* [4] verfügbar.

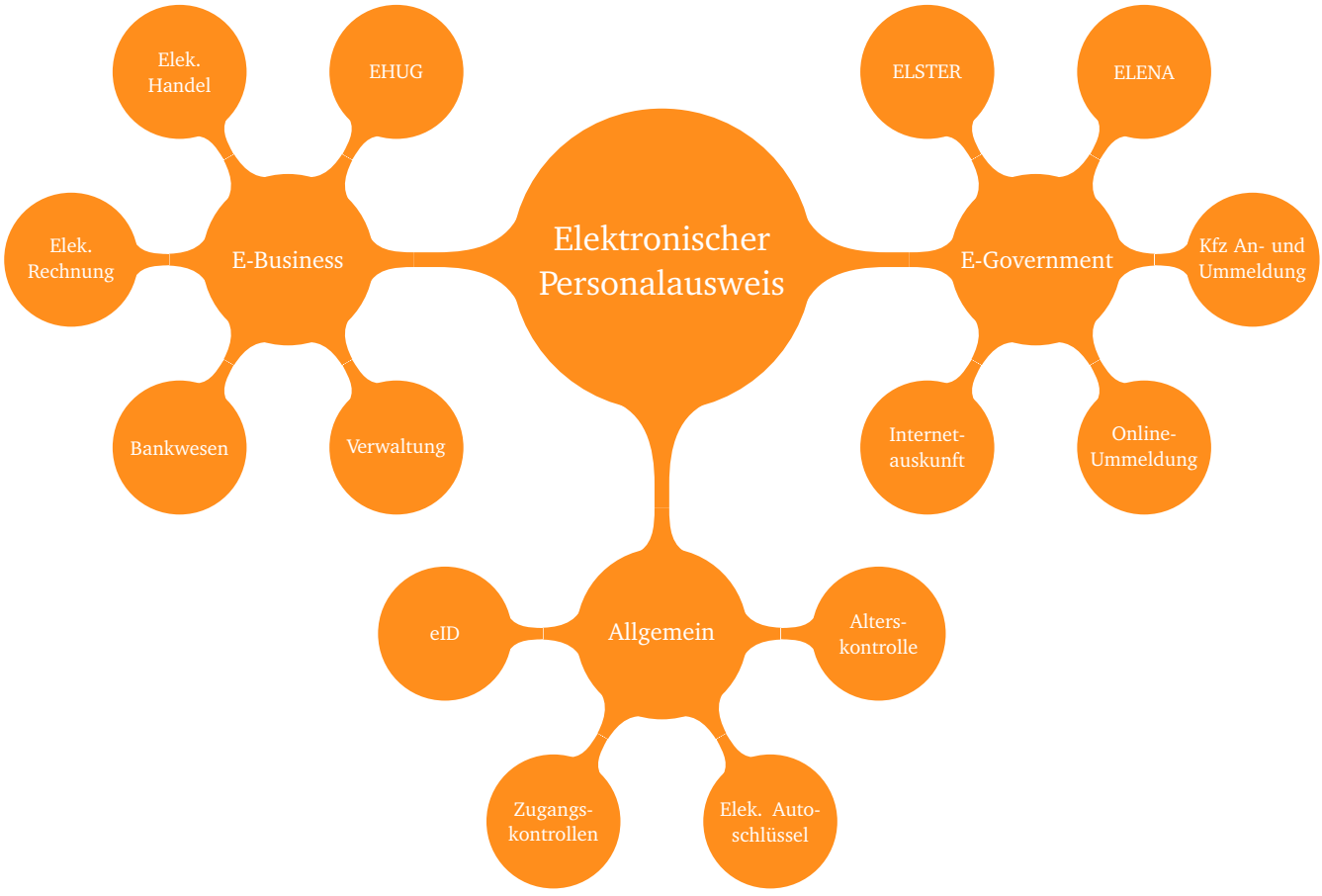

Abbildung 2.7.: Anwendungsgebiete des elektronischen Personalausweises

3 Schlüsselaustausch

Grundlage jeder symmetrisch verschlüsselten Verbindung ist ein gemeinsamer Schlüssel, der sowohl zum Verschlüsseln als auch zum Entschlüsseln verwendet wird. Das impliziert jedoch eine Reihe von Problemen: Verwenden mehrere Teilnehmer den gleichen symmetrischen Schlüssel, kann nicht eindeutig bestimmt werden, von wem die Nachricht ver- bzw. entschlüsselt wurde. Zusätzlich steigt die Gefahr, dass der Schlüssel öffentlich wird, je mehr Teilnehmern der Schlüssel zugänglich ist. Diese Nachteile werden durch asymmetrische Kryptosysteme kompensiert. Bei asymmetrischen Verschlüsselungen [28] wird zwischen einem öffentlichen Schlüssel (Public Key) zum Verschlüsseln und einem privaten Schlüssel (Private Key) zum Entschlüsseln differenziert. Jeder Teilnehmer verfügt über einen öffentlichen und einen privaten Schlüssel. Im Falle einer verschlüsselten Kommunikation verwendet der Absender den öffentlichen Schlüssel des Empfängers und dieser seinen privaten Schlüssel zum Entschlüsseln der Nachricht. Das bekannteste asymmetrische Kryptosystem wurde von R.L. Rivest, A. Shamir und L. Adleman im Jahre 1978 publiziert und ist unter dem Namen RSA [29] bekannt.

Der Austausch eines Schlüssels bei einer symmetrischen Verschlüsselung zwischen zwei Kommunikationsteilnehmern stellt jedoch ein Problem dar. Der Schlüssel bzw. Informationen zur Berechnung eines gemeinsamen Schlüssels können nur in unverschlüsselter Form übertragen werden, es ist also zu beachten, dass eine dritte Partei in der Lage wäre, diese Information ebenfalls zu erhalten und das Schutzziel der Vertraulichkeit der zu übertragenen Daten wäre gefährdet. Daraus folgt, dass der Schlüssel auf einem sicheren Weg ausgetauscht werden muss, zum Beispiel postalisch, oder die Information zur Schlüsselberechnung einem Dritten keine Möglichkeit geben den Schlüssel zu berechnen. Bei der asymmetrischen Verschlüsselung besteht diese Problematik nicht. Darüber hinaus besitzen diese Schlüssel eine deutlich höhere Gültigkeitsdauer, die in der Regel bis zu zwei Jahren betragen kann.

Bei der verschlüsselten Verbindung im Umfeld des Online-Banking ist es hinreichend, wenn für jede Sitzung ein Schlüssel mit dem Server der Bank und nicht ein Schlüssel für die gesamte Dauer der Kundenbeziehung vereinbart wird. Das heißt, dass ein Großteil der Schlüssel nur eine temporäre Verwendung findet. Im Internet findet ein Schlüsselaustausch zum Beispiel bei dem Aufbau einer verschlüsselten Verbindung über das Transport Layer Security (TLS) Protokoll statt. Einige Protokolle zum Schlüsselaustausch über Netzwerke basieren auf einer zentralen Stelle, die die Verwaltung und Zuordnung der Schlüssel für alle Teilnehmer organisiert. Laut ipoque GmbH [30] betrug der Anteil an verschlüsselten HTTP Übertragungen (HTTPS [31]) gerade mal 0,32 % des ausgewerteten Internet Verkehrs von 3 Petabyte, was immer noch 9,6 Terabyte entspricht. Würden alle Schlüssel für diese Verbindungen von einer zentralen Stelle verwaltet, hätte das eine immense organisatorische Herausforderung zur Folge, die technisch kaum zu realisieren wäre. Die Bündelung der Informationen an einer zentralen Stelle wäre ebenfalls mit der Problematik eines „Single Point of Failure“ (dt.: „einzelne Fehlerstelle“) verbunden.

> *„Key establishment is a process or protocol whereby a shared secret becomes available to two or more parties, for subsequent cryptographic use.“* [32]

Ein Protokoll zum Schlüsselaustausch hat demnach die Aufgabe einen Schlüssel zwei oder mehreren Parteien für eine verschlüsselte Kommunikation zur Verfügung zu stellen. Dabei sind mehrere Eigenschaften zu erfüllen:

- Durch Aufzeichnen einer verschlüsselten Kommunikation darf ein Angreifer nicht in der Lage sein, den verwendeten Schlüssel aus den ausgetauschten Nachrichten zu extrahieren.
- Ein Angreifer sollte beim Erlangen des Schlüssels die bereits getätigten Kommunikationen nicht entschlüsseln können. Daher wird bei jeder Sitzung bzw. Protokollablauf ein neuer Schlüssel berechnet und dieser wenn möglich kein zweites Mal verwendet. Diese Eigenschaft wird als *Forward Security* [33] bezeichnet und bedeutet, dass der Schaden durch das Erlagen von vertraulichen Informationen seitens unbefugter Personen möglichst gering gehalten wird.
- Das Ausprobieren aller möglichen Schlüssel sollte erschwert werden, indem beispielsweise nur ein Schlüssel pro Protokollablauf ausprobiert werden kann. Im Falle von Schlüsseln mit einer geringen Anzahl von Zeichen oder Ziffern sollte mit geeigneten Maßnahmen, wie einer Verzögerung oder Blockierung des Schlüssels, entgegen gewirkt werden.

3.1 Diffie-Hellman

Das im Jahre 1976 von Martin Hellman, Whitfield Diffie und Ralph Merkle publizierte Diffie-Hellman (DH) Protokoll [34] bildet die Grundlage für viele weitere Protokolle, die im Kern auf diesem aufbauen. Die Innovation des Diffie-Hellman Protokoll liegt im ausschließlichen Übertragen von Informationen zur Schlüsselgenerierung zwischen zwei Kommunikationsteilnehmern, der Schlüssel wird nicht direkt übertragen. Das Diffie-Hellman Protokoll basiert auf dem diskreten Logarithmus Problem, d. h. für die gegebenen Werte n,g und p ist es sehr schwer ein k zu finden, sodass $n = g^k \bmod p$ gilt [35]. Weitere Details zum diskreten Logarithmus Problem und dem konkreten Diffie-Hellman Problem werden unter anderem in dem Buch *Handbook of Applied Cryptography* [32] bereitgestellt. Der Protokollablauf ist in Abbildung 3.1 dargestellt.
Zu Beginn des Protokolls vereinbaren beide Teilnehmer die Parameter p und g, welche auf unsicherem Wege ausgetauscht werden können und infolgedessen öffentlich sind. Der erste Teilnehmer wählt eine zufällige Zahl a, berechnet $A = g^a \bmod p$ und sendet A an den zweiten Teilnehmer. Dieser wählt ebenfalls eine Zufallszahl b, berechnet $B = g^b \bmod p$ und antwortet mit B. Beide Kommunikationsteilnehmer berechnen dann den gemeinsamen Schlüssel K. Durch folgende Umformung sei verdeutlicht, dass beide Teilnehmer den gleichen Schlüssel verwenden:

$$K = A^b \bmod p \equiv (g^a \bmod p)^b \equiv g^{ab} \bmod p \equiv (g^b \bmod p)^a \equiv B^a \bmod p \text{ [34]}$$

Die Zufallszahlen a und b sind die privaten Schlüssel beider Teilnehmer und dementsprechend zu schützen. Für einen Angreifer sind zwar die Parameter p und g, sowie die Werte von A und B ersichtlich, aber ist es praktisch nicht möglich, in akzeptabler Zeit das diskrete Logarithmus-Problem zu lösen, das heißt z.B. ein a zu finden, für das $A = g^a \bmod p$ gilt. Das Diffie-Hellman Protokoll weist jedoch eine Schwäche auf: Ein Angreifer ist zwar nicht in der Lage aus den Informationen den Schlüssel zu berechnen, jedoch kann er sich zwischen zwei Kommunikationsteilnehmer postieren und als Vermittlungsstelle fungieren, was es ihm ermöglicht jegliche Kommunikation einzusehen und zu manipulieren. Man spricht dann vom Man-in-the-Middle-Angriff, den wir in Abschnitt 4.3.1 erläutern werden.

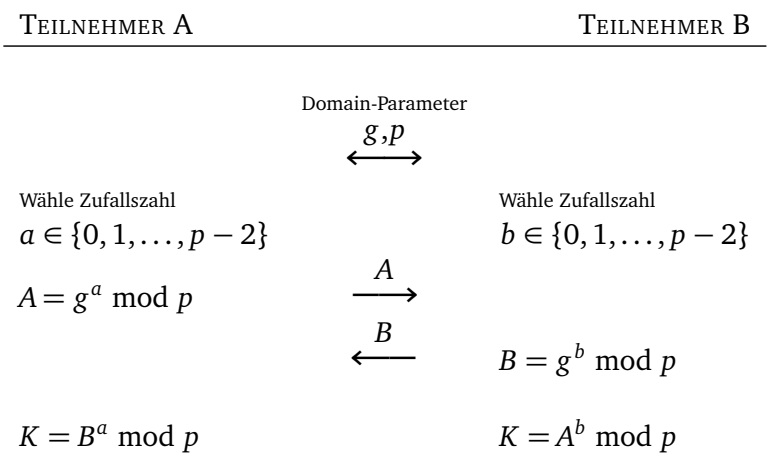

Abbildung 3.1.: Diffie-Hellman [34]

3.2 Elliptic Curve Diffie-Hellman

Der Einsatz eines Kryptosystems basierend auf elliptischen Kurven (ECC - Elliptic Curve Cryptosystem) ist insbesondere für Systeme mit geringem Speicher wie beispielsweise einer Chipkarte interessant. Im Vergleich zum Faktorisierungsproblem bzw. diskreten Logarithmus-Problem, welche bei RSA und dem in Kapitel 3.1 vorgestellten Diffie-Hellman Verfahren zum Einsatz kommen, ist die Berechnung des diskreten Logarithmus bei elliptischen Kurven deutlich schwieriger. Um die „gleiche“ Sicherheit zu erreichen, sind bei Verfahren basierend auf elliptische Kurven deutlich kleinere Schlüssel ausreichend. Abbildung 3.2 vergleicht die benötigte Schlüssellänge bei dem diskreten Logarithmus-Problem in elliptischen Kurven (ECDLP - Elliptic Curve Discrete Logarithm Problem) und dem klassischen diskreten Logarithmus-Problem (DLP - Discrete Logarithm Problem).

ECDLP	DLP
112 Bits	512 Bits
160 Bits	1024 Bits
224 Bits	2048 Bits
256 Bits	3072 Bits
384 Bits	7680 Bits
512 Bits	15360 Bits

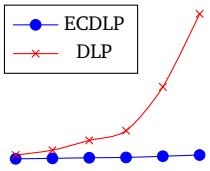

Abbildung 3.2.: Vergleich der Schlüssellänge [36]

Würde das RSA Verfahren bei Chipkarten wie dem ePA zum Einsatz kommen, müssten erheblich größere Speicherkapazitäten im Chip integriert werden. Das Diffie-Hellman Protokoll lässt sich modifizieren, sodass die Berechnungen bzw. Schlüssel auf elliptischen Kurven basieren. Zu Beginn des Elliptic Curve Diffie-Hellman (ECDH) Protokolls werden die Domain Parameter ausgetauscht. Die Parameter seien dabei wie folgt definiert: Sei p eine Primzahl des endlichen Körpers $\mathbb{F}_p$, $E(\mathbb{F}_p)$ eine elliptische Kurve, G ein Basispunkt auf der Kurve (weitere Details siehe [37]). Zu Beginn des Protokolls werden die Domain-Parameter p, a, b, G, n, h ausgetauscht, dann wählen beide Teilnehmer einen zufälligen privaten Schlüssel SK, berechnen einen Punkt PK (der als öffentlicher Schlüssel betrachtet werden kann) auf der Kurve $E(\mathbb{F}_p)$ und tauschen diesen aus. Im letzten Schritt wird der gemeinsame Schlüssel K berechnet bestehend aus dem eigenen privaten Schlüssel SK und dem öffentlichen Schlüssel PK des anderen Teilnehmers.

TEILNEHMER A		TEILNEHMER B
	Domain-Parameter p,a,b,G,n,h $\longleftrightarrow$	
Wähle zufälligen privaten Schlüssel SK_A		Wähle zufälligen privaten Schlüssel SK_B
$PK_A = SK_A * G$	$PK_A \longrightarrow$	
	$PK_B \longleftarrow$	$PK_B = SK_B * G$
Berechne gemeinsamen Schlüssel $K = SK_A * PK_B$		Berechne gemeinsamen Schlüssel $K = SK_B * PK_A$

$$K = SK_A * PK_B = SK_A * (SK_B * G) = SK_B * (SK_A * G) = SK_B * PK_A$$

Abbildung 3.3.: Elliptic Curve Diffie-Hellman [38]

3.3 Passwort-basierter Schlüsselaustausch

Das vorgestellte Diffie-Hellman Verfahren ermöglicht einen Schlüsselaustausch zwischen zwei Parteien ohne Informationen zu übertragen, die es einem Dritten ermöglichen, den Schlüssel zu berechnen. Der gravierende Nachteil ist jedoch die fehlende Authentifizierung der Teilnehmer. Nach einem erfolgreichen Diffie-Hellman Schlüsselaustausch können sich beide Teilnehmer auf eine vertrauliche Kommunikation verlassen, es ist aber nicht offensichtlich, mit wem sie kommunizieren. Erforderlich ist demnach nicht nur eine verschlüsselte Kommunikation, sondern auch eine Authentifizierung der Teilnehmer.
Mittels digitaler Zertifikate könnte eine zuverlässige Authentifizierung der jeweiligen Parteien realisiert werden. Dies setzt jedoch voraus, dass die Teilnehmer die Zertifikate überprüfen können und die Übertragung der Zertifikate nur in unverschlüsselter Form geschehen kann. Dadurch entsteht ein Sicherheitsrisiko, weil eine dritte Partei die Identitäten anhand der Zertifikate feststellen kann und Bewegungsprofile erstellt werden können. Auch wenn einem Dritten die Berechnung des Schlüssels nicht möglich wäre, ist dieses jedoch eine Verletzung des Schutzziels der Vertraulichkeit.
Passwort-basierte Verfahren bieten eine Alternative zur sicheren und vertrauenswürdigen Authentifizierung der Teilnehmer. Auf Grundlage eines gemeinsamen Passwortes, das beiden Teilnehmern bekannt sein muss, bildet dieses die Basis des Schlüsselaustauschs. Es entsteht eine Kopplung des Passwortes mit dem Schlüssel, der für die vertrauliche Kommunikation verwendet wird. Diese Verfahren verfolgen das gleiche Ziel wie klassische Protokolle zum Schlüsselaustausch, d. h. die Berechnung eines Sitzungsschlüssels zwischen zwei Parteien. Ein Passwort-basiertes Protokoll lässt sich demnach als ein Verfahren zum Schlüsselaustausch definieren, das anhand eines vorher bekannten Geheimnisses eine Authentifizierung der Teilnehmer ermöglicht und das Geheimnis indirekt in die Berechnung des Sitzungsschlüssels mit einfließt.
Das dezentrale Speichern der Passwörter ist ebenfalls als ein Vorteil dieser Protokolle zu sehen. Die Gültigkeit eines Passwortes kann dabei autonom geprüft werden ohne einen Abgleich mit einer zentralen Stelle, die die Passwörter verwaltet. Die Entropie des Passwortes ist dabei flexibel; in einem Szenarien in denen sich Personen authentifizieren müssen werden jedoch Passwörter mit einer geringen Ziffern- bzw. Zeichenanzahl verwendet.

Unter anderem existieren folgende passwort-basierte Protokolle:

- Encrypted Key Exchange (EKE) [39]
- Augmented-Encrypted Key Exchange (A-EKE) [40]
- Diffie-Hellman Encrypted Key Exchange (DH-EKE) [41]
- Password Authenticated Key Exchange (PAK) [42]
- Open Key Exchange (OKE) [43]

Eine Übersicht über mehrere passwort-basierte Verfahren ist in dem Artikel [44] von David P. Jablon verfügbar sowie in [41] und [45]. Der Ablauf dieser Protokolle ist sehr ähnlich. Zu Beginn wird eine Zufallszahl mit dem gemeinsamen Passwort verschlüsselt und ausgetauscht oder es wird eine Prüfsumme anhand weiterer Informationen und dem Passwort erstellt. Nur wenn beide Teilnehmer das Passwort kennen, wird derselbe Schlüssel für die weitere Kommunikation berechnet. Viele dieser Protokolle sind jedoch patentiert und ungeeignet für elliptische Kurven.

3.3.1 Simple Password Exponential Key Exchange

Als eines der klassischen passwort-basierten Protokolle stellen wir das Simple Password Exponential Key Exchange (SPEKE) Protokoll vor, insbesondere wegen der Ähnlichkeit und den Parallelen zum PACE bzw. PASC Protokoll, die in Kapitel 3.3.2 bzw. 3.3.3 folgen. Das SPEKE Protokoll lässt sich in zwei Phasen unterteilen: In der Ersten (Abb. 3.4 ⟨1⟩ - ⟨2⟩) werden Informationen zur Berechnung des gemeinsamen Schlüssels ausgetauscht und der Schlüssel berechnet, ähnlich dem Diffie-Hellman Protokoll. Die zweite Phase (Abb. 3.4 ⟨3⟩) dient der gegenseitigen Authentifizierung (ISO/IEC 9798 [46]), d. h. es wird überprüft, ob beide Teilnehmer den gleichen gemeinsamen Schlüssel besitzen und somit gewährleistet ist, dass beide das Passwort π kennen. Die Authentifizierung erfolgt durch das Verschlüsseln von Zufallszahlen und der Überprüfung, ob die andere Partei diese erfolgreich ver- bzw. entschlüsseln kann. Das Patent von SPEKE beschreibt das Protokoll generischer als in Abb. 3.4 skizziert. Anstatt der Hash-Funktion $\mathscr{H}$ wird nur eine nicht weiter definierte Funktion f angegeben (d. h. $X = f(\pi)^x$).

	Teilnehmer A		Teilnehmer B
⟨1⟩	Wähle Zufallszahl x		
	$X = (\mathscr{H}(\pi)^2)^x$	$\xrightarrow{X}$	
			Wähle Zufallszahl y
		$\xleftarrow{Y}$	$Y = (\mathscr{H}(\pi)^2)^y$
⟨2⟩	$K_\pi = \mathscr{H}(Y)^x$		$K_\pi = \mathscr{H}(X)^y$
⟨3⟩	Wähle Zufallszahl r_A		
	$m_1 = E(K_\pi, r_A)$	$\xrightarrow{m_1}$	
			Wähle Zufallszahl r_B
		$\xleftarrow{m_2}$	$m_2 = E(K_\pi, r_A, r_B)$
	Entschlüssel m_2 und prüfe r_A		
	$m_3 = E(K_\pi, r_B)$	$\xrightarrow{m_3}$	Entschlüssel m_3 und prüfe r_B

Abbildung 3.4.: SPEKE [41] [47]

SPEKE: Protokollablauf

1. Teilnehmer A quadriert den Hash-Wert des Passwortes π und verwendet eine Exponentialfunktion. Das Ergebnis wird an Teilnehmer B gesendet, der analog mit seiner Zufallszahl y verfährt.
2. Beide Teilnehmer berechnen den gemeinsamen Schlüssel K_π durch Anwendung einer Hash-Funktion über das Ergebnis des anderen und einer Exponentialfunktion.
3. Teilnehmer A wählt eine Zufallszahl r_A, verschlüsselt und sendet diese an B. Dieser entschlüsselt die Nachricht m_1 und extrahiert r_A. Teilnehmer B wählt ebenfalls eine Zufallszahl r_B und verschlüsselt diese zusammen mit r_A zu einer Nachricht m_2. Diese wird von A entschlüsselt und die Zufallszahl r_A überprüft. Wenn diese nicht übereinstimmt, dann verfügt Teilnehmer B nicht über den gleichen Schlüssel K_π und das Protokoll wird beendet. Ansonsten extrahiert Teilnehmer A die Zufallszahl r_B und schickt diese verschlüsselt zurück an Teilnehmer B. Dieser prüft analog r_B.

3.3.2 Password Authenticated Connection Establishment

Die Sicherheit, das heißt die Gewährleistung der Schutzziele Vertraulichkeit, Verfügbarkeit, Integrität und Authentizität, der auf dem elektronischen Personalausweis digital gespeicherten Daten, ist eine große Herausforderung. Um den Sicherheitsanforderungen der sensiblen Daten, bestehend aus den biometrischen Gesichts- und Fingerabdruckdaten, gerecht zu werden, muss sichergestellt werden, dass nur authentifizierte Lesegeräte Zugriff erhalten und Identitätsdiebstahl durch kopieren der Daten aufgedeckt werden kann. Diese Bedingungen werden durch Extended Access Control (EAC) [3], bestehend PACE, Chip-Authentifizierung und Terminal-Authentifizierung, erfüllt. Bevor Chip- und Terminal-Authentifizierung zum Einsatz kommen, bedarf es der Zustimmung des Ausweisinhabers und einer verschlüsselten und integeren Verbindung zwischen Lesegerät und dem elektronischen Personalausweis. Diese essenziellen Anforderungen realisiert das Password Authenticated Connection Establishment (PACE) Protokoll. Die Entwicklung von PACE sollte die Vorteile einer kontaktlosen Chipkarte nutzbar machen und die Nachteile kompensieren. Die Ziele von PACE sind:

- Kontrolle des optischen Zugriffs des Lesegerätes
- Starke und sichere Sitzungsschlüssel für die Kommunikation

Die erste Aufgabe von PACE ist es, sicherzustellen, dass ein Lesegerät optischen Zugang zum elektronischen Personalausweis hat und somit keine ungewollte und vom Inhaber unbemerkte Kommunikation statt findet. Durch die Eigenschaft einer kontaktlosen Chipkarte des elektronischen Personalausweises, ist eine Kommunikation auch über eine kurze Distanz möglich. Dadurch könnte — vom Inhaber unbemerkt — eine Kommunikation mit dem elektronischen Personalausweis aufgebaut werden. Dies wird durch das PIN und PUK Konzept des ePA realisiert (siehe Kapitel 2.2). Ein Lesegerät bestätigt einen optischen Zugriff auf den elektronischen Personalausweis durch die auf dem Ausweis aufgedruckte Card Access Number (CAN) oder durch eine vom Inhaber am Lesegerät eingegebene PIN. Im letzten Schritt des PACE Protokolls erfolgt eine Verifikation der Schlüssel. Kann das Lesegerät kein korrektes Passwort nachweisen, wird das Protokoll abgebrochen. Der Inhaber ist damit in der Lage den Zugriff auf seinen elektronischen Personalausweis zu kontrollieren, in dem er zuerst seine geheime PIN eingeben muss oder dem Lesegerät optischen Zugriff auf die CAN gewährt, bevor eine Kommunikation zwischen Lesegerät und elektronischem Personalausweis zustande kommt. Bei klassischen Magnetstreifenkarten wird diese Einwilligung durch das Einführen der Karte in das Lesegerät realisiert.
Die zweite Aufgabe von PACE besteht darin, starke und sichere Schlüssel für eine Verbindung bereitzustellen. Da die CAN sowie die geheime PIN des Inhabers nur eine geringe Ziffernanzahl aufweisen und demnach eine geringe Entropie besitzen, sind diese als Schlüssel für eine sichere Kommunikation ungeeignet. Würde auf PACE verzichtet werden und für die verschlüsselte Kommunikation bei Extended Access Control nur die CAN oder die PIN des Inhabers zum Einsatz kommen, wäre eine sehr einfache Verschlüsselung die Folge. Weil davon auszugehen ist, dass die Kommunikation zwischen Lesegerät und dem elektronischen Personalausweis aufgezeichnet und gespeichert werden kann, besteht die Möglichkeit die verschlüsselten Daten zu einem späteren Zeitpunkt und in einem anderen technischen Umfeld anzugreifen. In Anbetracht der Tatsache, dass die Gültigkeit des elektronischen Personalausweises zehn Jahre beträgt, wäre eine Verschlüsselung auf Grundlage der einzelnen Passwörter wie CAN, PIN usw. (siehe Kapitel 2.6) hinfällig. Die Möglichkeiten zur Steigerung der Komplexität der Passwörter, beispielsweise durch eine höhere Anzahl der Ziffern der CAN oder der PIN des Inhabers, sind erschöpft, weil die aufgedruckte CAN sowieso leicht zugänglich ist. Eine PIN des Inhabers mit beispielsweise acht Ziffern würde den Grad der Verschlüsselung nur marginal erhöhen und eher eine geringere Akzeptanz bei Bürgerinnen und Bürger implizieren, sowie die Verwendbarkeit des elektronischen Personalausweises durch Sperrung in Folge von mehrmaliger Falscheingabe der PIN negativ beeinflussen. Das PACE Protokoll hat also die Aufgabe diese sehr unsicheren Schlüssel zu ersetzen und eine Verschlüsselung auf hohem Niveau mittels eines komplexen Schlüssels zu etablieren. Die Verschlüsselung dient der Absicherung der Luftschnittstelle zwischen dem elektronischen Personalausweis und einem Lesegerät und wegen

der Möglichkeit einer Aufzeichnung der Kommunikation muss sie über viele Jahre einen hinreichenden Schutz der Vertraulichkeit bieten. Es ist also sicherzustellen, dass die Verschlüsselung zu einem späteren Zeitpunkt nicht gebrochen werden kann. Solche Angriffe werden „Offline-Angriffe“ genannt, weil die Kommunikation bereits stattgefunden hat, die verschlüsselten Daten jedoch immer noch vorliegen und angegriffen werden können.

PACE: Protokollablauf

Abbildung 3.5 illustriert den Protokollablauf, die einzelnen Schritt werden im Folgenden erläutert:

1. Der Chip des elektronischen Personalausweises generiert eine Zufallszahl s, verschlüsselt diese mit dem gemeinsamen Schlüssel K_π und schickt sie an das Lesegerät. Die Domain-Parametern D_{PICC} extrahiert das Lesegerät aus der Datei `EF.CardAccess`, die zuvor ausgelesen werden muss. Das Lesegerät entschlüsselt die Nachricht z mit Hilfe des Schlüssel K_π und erhält somit die vom ePA berechnete Zufallszahl s. Der Schlüssel K_π wird zum Beispiel aus der Personal Identification Number (PIN) des Ausweis-Inhaber berechnet.

2. Die Teilnehmer tauschen weitere Daten zur Berechnung der Diffie-Hellman Domain Parameter aus und berechnen den neuen Punkt G'.

3. Der elektronische Personalausweis und das Lesegerät berechnen zufällige und flüchtige Diffie-Hellman Schlüssel, die sich aus einem privaten Schlüssel $\widetilde{SK_{PICC}}$ bzw. $\widetilde{SK_{PCD}}$ und einem öffentlichen Schlüssel $\widetilde{PK_{PICC}}$ bzw. $\widetilde{PK_{PCD}}$ zusammensetzen.

4. Beide Kommunikationspartner berechnen den gemeinsamen Schlüssel K bestehend aus den privaten und öffentlichen Schlüssel und den Domain Parametern. Aus dem Schlüssel K leiten beide einen Schlüssel $K_{Enc} = KDF_{Enc}(K, [r])$ zur Verschlüsselung und einen Schlüssel $K_{MAC} = KDF_{MAC}(K, [r])$ für die Authentifizierung ab. Die Key Derivation Function (KDF) dient dazu aus einem geheimen Schlüssel K weitere Schlüssel abzuleiten, die Zufallszahl r dient zur Steigerung der Entropie.

5. Der ePA und das Lesegerät generieren jeweils einen Authentisierungstoken T_{PICC} bzw. T_{PCD}, tauschen diesen aus und verifizieren ihn gegenseitig. T_{PICC} bzw. T_{PCD} ist eine Prüfsumme basierend auf dem Schlüssel K_{MAC} des öffentlichen Schlüssels des Lesegerätes bzw. des ePA.

Mit einer erfolgreichen Durchführung des Password Authenticated Connection Establishment Protokoll startet die sichere Kommunikation zwischen dem elektronischen Personalausweis und einem Lesegerät. Als Sitzungsschlüssel werden K_{MAC} zum Authentifizieren und K_{Enc} zum Ver- bzw. Entschlüsseln verwendet. Die in Abbildung 3.5 verwendeten Abkürzung sind in Tabelle 3.6 erläutert.

	Proximity Integrated Circuit Chip		Proximity Coupling Device
⟨1⟩	Wähle Zufallszahl s		
	$z = E(K_\pi, s)$	$\xrightarrow{D_{PICC}, z}$	$s = D(K_\pi, z)$
⟨2⟩	Wähle Zufallszahl x		
	$X = x * G$	$\xrightarrow{X}$	
			Wähle Zufallszahl y
		$\xleftarrow{Y}$	$Y = y * G$
	$H = x * Y$		$H = y * X$
	$G' = s * G + H$		$G' = s * G + H$
⟨3⟩	Berechne Schlüsselpaar $\widetilde{SK_{PICC}}, \widetilde{PK_{PICC}}$		
	$\widetilde{PK_{PICC}} = \widetilde{SK_{PICC}} * G'$	$\xrightarrow{\widetilde{PK_{PICC}}}$	
			Berechne Schlüsselpaar $\widetilde{SK_{PCD}}, \widetilde{PK_{PCD}}$
		$\xleftarrow{\widetilde{PK_{PCD}}}$	$\widetilde{PK_{PCD}} = \widetilde{SK_{PCD}} * G'$
⟨4⟩	$K = \widetilde{SK_{PICC}} * \widetilde{PK_{PCD}} * G'$		$K = \widetilde{SK_{PICC}} * \widetilde{PK_{PCD}} * G'$
⟨5⟩	$T_{PICC} = MAC(K_{MAC}, \widetilde{PK_{PCD}})$	$\xrightarrow{T_{PICC}}$	
		$\xleftarrow{T_{PCD}}$	$T_{PCD} = MAC(K_{MAC}, \widetilde{PK_{PICC}})$

Abbildung 3.5.: PACE [3, Kapitel 4.2]

Name	Abkürzung
Chip Authentication Public Key	PK_{PICC}
Chip Authentication Private Key	SK_{PICC}
Domain Parameters	D
Ephemeral Public Key	$\widetilde{PK}$
Ephemeral Private Key	$\widetilde{SK}$
Key Derivation Function	**KDF**
Proximity Integrated Circuit Chip	PICC
Proximity Coupling Device	PCD
Terminal Authentication Public Key	PK_{PCD}
Terminal Authentication Private Key	SK_{PCD}

Abbildung 3.6.: PACE Legende [3]

3.3.3 Password Authenticated Secure Channel

Das Password Authenticated Secure Channel[1] (PASC) Protokoll wird von der Firma Gemalto[2] entwickelt und stellt eine Alternative zu dem PACE Protokoll dar. Es ist ebenfalls ein passwort-basiertes Schlüsselaustausch Protokoll, setzt jedoch vollständig auf ECDH, d. h. auf ein Diffie-Hellman Verfahren mit elliptischen Kurven. Analog zu PACE ist das Ziel eine starke, verschlüsselte Kommunikation durch einen entropiereichen Schlüssel zwischen zwei Parteien aufzubauen. Grundlage ist ebenfalls ein schwacher gemeinsamer Schlüssel, beispielsweise eine PIN, der sowohl dem Lesegerät als auch dem Chip bekannt sein muss.

PASC: Protokollablauf

Abbildung 3.7 illustriert den Protokollablauf, die einzelnen Schritt werden im Folgenden erläutert:

1. Der Chip generiert eine Zufallszahl s, verschlüsselt diese mit dem gemeinsamen Schlüssel π und sendet diese an das Lesegerät. Das Lesegerät entschlüsselt die Nachricht mittels des gemeinsamen Schlüssels π.

2. Der Chip wählt wieder eine Zufallszahl x und sendet den Hash-Wert x' an das Lesegerät. Das Lesegerät generiert ebenfalls eine Zufallszahl und sendet sie an den Chip. Nach dem Empfang sendet der Chip die Zufallszahl x. Das Lesegerät besitzt jetzt die Zufallszahl x und den vom Chip berechneten Hash-Wert dieser Zufallszahl. Dieser wird überprüft und bei keiner Übereinstimmung das Protokoll abgebrochen. Andernfalls berechnen beide Kommunikationsteilnehmer G', sowie H mittels der Funktion *Hash2Point*, die eine Zeichenfolge einem Punkt auf der elliptischen Kurve zuordnet (siehe [48, Seite 3]).

3. Der Chip wählt einen zufälligen privaten Schlüssel $\widetilde{SK_{PICC}}$ und berechnet $\widetilde{PK_{PICC}}$. Das Lesegerät handelt analog und berechnet und versendet $\widetilde{PK_{PCD}}$. Beide Teilnehmer berechnen Z.

4. Beide Teilnehmer konkatenieren Z, $\widetilde{PK_{PICC}}$, $\widetilde{PK_{PCD}}$ zum Schlüssel K und leiten mittels der Key Derivation Function (KDF) die Schlüssel K_{ENC} und K_{MAC} aus K ab.

5. Der folgende Schritt, das Signieren der Schlüssel, ist bei PASC optional. Der Chip berechnet einen Hash-Wert von $\widetilde{PK_{PCD}}$ (unter Verwendung des Schlüssel K_{MAC}) und sendet ihn an das Lesegerät. Das Lesegerät verfährt analog und berechnet und sendet den Hash-Wert von $\widetilde{PK_{PICC}}$. Beide Parteien überprüfen daraufhin die gegenseitig generierten Hash-Werte.

6. Das Lesegerät und der Chip berechnen die Sitzungsschlüssel $State_{ENC}$ und $State_{MAC}$ mittels einer Hash-Funktion über die Konkatenation von $\widetilde{PK_{PCD}}$ und $\widetilde{PK_{PICC}}$. Diese Schlüssel werden dann für die weitere verschlüsselte Kommunikation, dem *Secure Messaging* eingesetzt.

[1] Zukünftig unter dem Namen PACE-EU bekannt.
[2] `http://www.gemalto.com`

	PROXIMITY INTEGRATED CIRCUIT CHIP		PROXIMITY COUPLING DEVICE
⟨1⟩	Wähle Zufallszahl s		
	$z = E(K_\pi, s)$	$\xrightarrow{D_{PICC}, z}$	$s = D(K_\pi, z)$
⟨2⟩	Wähle Zufallszahl x		
	$x' = \mathcal{H}_1(x)$	$\xrightarrow{x'}$	
			Wähle Zufallszahl y
		$\xleftarrow{y}$	
		$\xrightarrow{x}$	$\mathcal{H}_1(x) \stackrel{?}{=} x'$
	$H = \text{Hash2Point}(x \parallel y)$ $G' = s * G + H$		$H = \text{Hash2Point}(x \parallel y)$ $G' = s * G + H$
⟨3⟩	Berechne Schlüsselpaar $\widetilde{SK_{PICC}}, \widetilde{PK_{PICC}}$		
	$\widetilde{PK_{PICC}} = \widetilde{SK_{PICC}} * G'$	$\xrightarrow{\widetilde{PK_{PICC}}}$	
			Berechne Schlüsselpaar $\widetilde{SK_{PCD}}, \widetilde{PK_{PCD}}$
		$\xleftarrow{\widetilde{PK_{PCD}}}$	$\widetilde{PK_{PCD}} = \widetilde{SK_{PCD}} * G'$
	$Z = \widetilde{PK_{PCD}} * \widetilde{SK_{PICC}}$		$Z = \widetilde{PK_{PICC}} * \widetilde{SK_{PCD}}$
⟨4⟩	$K = Z \parallel \widetilde{PK_{PICC}} \parallel \widetilde{PK_{PCD}}$		$K = Z \parallel \widetilde{PK_{PICC}} \parallel \widetilde{PK_{PCD}}$
⟨5⟩	$T_{PICC} = MAC(K_{MAC}, \widetilde{PK_{PCD}})$	$\xrightarrow{T_{PICC}}$	
		$\xleftarrow{T_{PCD}}$	$T_{PCD} = MAC(K_{MAC}, \widetilde{PK_{PICC}})$
⟨6⟩	$\text{State}_{ENC} = \mathcal{H}_2(\widetilde{PK_{PICC}} \parallel \widetilde{PK_{PCD}})$		$\text{State}_{ENC} = \mathcal{H}_2(\widetilde{PK_{PICC}} \parallel \widetilde{PK_{PCD}})$
	$\text{State}_{MAC} = \mathcal{H}_3(\widetilde{PK_{PICC}} \parallel \widetilde{PK_{PCD}})$		$\text{State}_{MAC} = \mathcal{H}_3(\widetilde{PK_{PICC}} \parallel \widetilde{PK_{PCD}})$

Abbildung 3.7.: PASC [48]

3.3.4 Zusammenfassung und Analyse

Das PACE und das PASC Protokoll verfolgen die gleichen Ziele (den Aufbau einer vertraulichen und integeren Verbindung zwischen Lesegeräte und elektronischem Personalausweis) und haben im Wesentlichen den gleichen Protokollablauf. Beides sind passwort-basierte Protokolle, die anhand des gemeinsamen Passwortes π eine gegenseitige Authentifizierung durchführen und starke Sitzungsschlüssel für die weitere Kommunikation berechnen.
Der Chip generiert eine Zufallszahl und verschlüsselt diese mit einem vom Passwort π abgeleiteten Schlüssel. Nur durch die Kenntnis dieses Passwortes kann das Lesegerät durch Entschlüsseln der Nachricht die Zufallszahl extrahieren. Der nächste Schritt, den beide Protokolle durchführen, ist der Austausch von Informationen für die Berechnung eines neuen Punktes G' auf der elliptischen Kurve. Dabei fließt die vom Chip generierte Zufallszahl in die Berechnung mit ein, was für eine Abhängigkeit des neuen Punktes G' an das verwendete Passwort π impliziert. Dadurch ist sichergestellt, dass der Punkt G' nur von einem Terminal berechnet werden kann, das Kenntnis über das Passwort hat. Beide Protokolle führen dann einen anonymen Diffie-Hellman Schlüsselaustausch durch, um einen gemeinsamen Schlüssel zu berechnen, der dann als Basis für die Sitzungsschlüssel K_{ENC} und K_{MAC} verwendet wird. Im letzten Schritt erfolgt gegebenenfalls eine Authentifizierung der beiden Teilnehmer durch „Signieren“ der öffentlichen Schlüssel.

Zusammenfassen lassen sich die Parallelen des PACE und PASC Protokolls auf den Austausch einer verschlüsselten Zufallszahl, Berechnung eines neuen Punktes auf der elliptischen Kurve und der Durchführung eines Diffie-Hellman zum Berechnen eines gemeinsamen Schlüssels. Die Authentifizierung der Schlüssel ist bei PASC optional.
Einer der Unterschiede zwischen den beiden Protokollen ist der Austausch der Informationen bzw. Zufallszahlen zur Berechnung des neuen Punktes auf der Kurve. Das PACE Protokoll setzt in diesem Punkt auf ein klassisches Diffie-Hellman Verfahren, wohingegen PASC einen anderen Ansatz verfolgt. Auffällig an der PASC-Variante ist die Berechnung des Hash-Wertes der Zufallszahl des Chips. Für die Analyse dieses Vorgehens betrachten wir zuerst die Möglichkeit, dass der Chip seine generierte Zufallszahl direkt an das Terminal sendet.

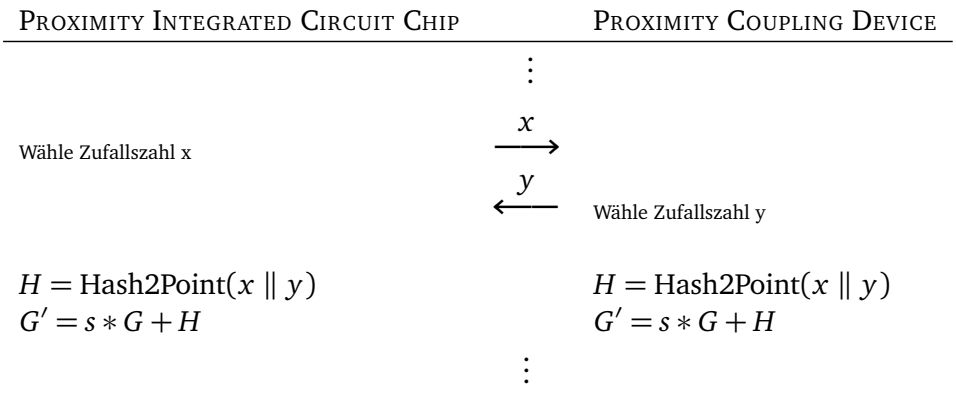

Abbildung 3.8.: PASC Protokoll-Analyse I

Die Berechnung von H hat direkte Auswirkungen auf den neu gewählten Punkt G' und auf diese Weise auch auf den gemeinsamen Schlüssel, der als letztes von beiden Teilnehmern berechnet wird. Würde PASC die in Abbildung 3.8 illustrierte Modifikation verwenden, könnte ein böswilliges Terminal seine Zufallszahl β so wählen, dass ein *günstiges* bzw. *schwaches* G' berechnet wird. In diesem Fall kann ein böswilliges Terminal indirekt die Berechnung des gemeinsamen Schlüssels beeinflussen. Um diesem

Sicherheitsproblem entgegenzuwirken, könnte die Reihenfolge des Austausches der Zufallszahlen angepasst werden. Dieses Szenario verdeutlicht Abbildung 3.9.

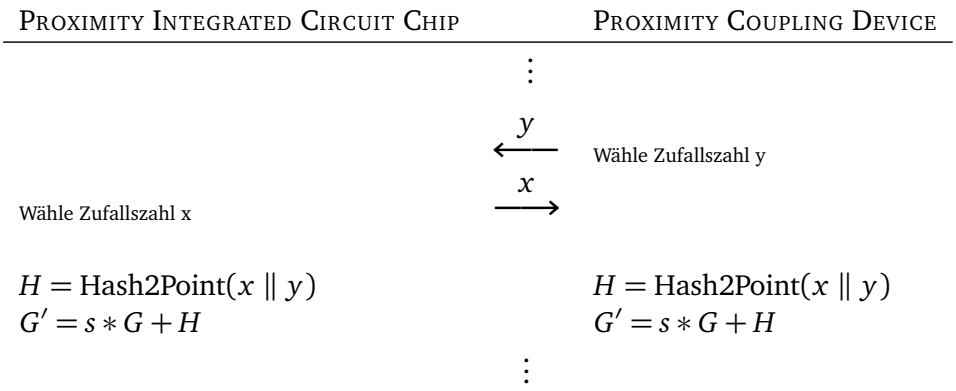

Abbildung 3.9.: PASC Protokoll-Analyse II

Die Problematik hätte sich jetzt zu Gunsten des Chips verlagert, das heißt, das Lesegerät müsste zuerst eine Zufallszahl wählen, bevor der Chip seine preisgibt. Dadurch würde jedoch das Sicherheitsrisiko nur verlagert – im Gegensatz zum böswilligen Terminal könnte jetzt ein manipulierter bzw. böswilliger Chip die Berechnung des gemeinsamen Schlüssels beeinflussen. Durch die Verwendung des Hash-Wertes der Zufallszahl x ist demnach sichergestellt, dass weder der Chip noch das Terminal die Berechnung des Schlüssels manipulieren bzw. beeinflussen können.
Der Chip sendet, in Form des Hash-Wertes, einen Repräsentanten der Zufallszahl, aus dem keine Informationen über die Zufallszahl selber zu entnehmen sind. Dieser Repräsentanten ermöglicht dem Terminal aber nach dem Erhalt der Zufallszahl die Zugehörigkeit dieser mit dem Repräsentanten zu prüfen. Unter der Annahme, dass die verwendete Hashfunktion kollisionsresistent ist und demnach jede Zufallszahl in einem gewählten Intervall einen anderen bzw. eindeutigen Hashwert besitzt sowie keine Schlussfolgerungen aus dem Hashwert auf die Zufallszahl gewonnen werden können, kann weder ein Terminal noch ein Chip Einfluss auf die Schlüsselberechnung nehmen. Der Chip und das Terminal sind demnach genötigt, eine Zufallszahl zu wählen, ohne die jeweils andere zu kennen und ggf. die eigene anzupassen. Die Gefahr bzw. Sicherheitsbedrohung, die aus einer direkten bzw. indirekten Manipulation der berechneten Schlüssel folgt, ist nicht zu unterschätzen. Die Sicherheit von PACE und PASC beruhen auf dem Passwort π und dem Punkt G' auf der elliptischen Kurve. Das Passwort π ist statisch und besitzt nur eine geringe Entropie; die Anzahl der Ziffern und der Zeichensatz (Alphabet) ist im Allgemeinen bekannt. Bei der PIN des elektronischen Personalausweises besteht das Alphabet aus sechs Ziffern von 0 bis 9. Der Suchraum bzw. Bereich aller möglichen Passwörter ist demnach bekannt und im Allgemeinen „sehr klein“. Durch ein Eingreifen in die Berechnung durch den Chip bzw. das Terminal ist der Punkt H nicht mehr dynamisch bzw. gleichverteilt. Ziehen wir noch den geringen Wertebereich von π hinzu, gelten diese Eigenschaften auch für den Punkt G'. Der Punkt wird daraufhin nicht mehr zufällig gewählt bzw. berechnet. Das Ziel eines böswilligen Chips bzw. Terminals wäre der Versuch, die Berechnung von H so zu beeinflussen, dass dieses auch bereits $\log_G H$ kennt. Anschließend lässt sich der diskrete Logarithmus von G' effizient berechnen und die Verschlüsselung wäre gebrochen [49].
Die abschließende Frage, die noch offen bleibt ist: Warum beginnt der Chip den Protokollschritt mit dem Senden des Hash-Wertes seiner Zufallszahl und nicht das Terminal? Die Entscheidung kann ggf. damit begründet werden, dass ein Chip allgemein als vertrauenswürdiger einzustufen ist als ein Lesegerät, weil eine Manipulation des Chips sicherlich aufwendiger ist und das klassische Angriffsszenario aus einer vom Inhaber unbemerkten Kommunikation zwischen Lesegerät und Ausweis besteht.

Der zweite Unterschied zwischen PACE und PASC zeigt sich in den Protokollabläufen in Abbildung 3.5 und 3.7 in Bezug auf die Authentisierungstoken. Die Spezifikation von Extended Access Control Version 2.01 [3] fordert die Berechnung einer Prüfsumme der öffentlichen Schlüssel $\widetilde{PK_{PICC}}$ und $\widetilde{PK_{PCD}}$. Für PASC ist dieser Schritt als optional vorgesehen. Die Frage, die sich stellt, ist, ob eine gegenseitige Authentifizierung mit Hilfe der Token T_{PICC} und T_{PCD} bei PACE bzw. PASC notwendig ist bzw. einen zusätzlichen Sicherheitsgewinn mit sich bringt?
Betrachten wir die Ausführung von PASC ohne die Verwendung von Authentisierungstoken: Wir nehmen an, dass das Lesegerät den Schlüssel π nicht kennt und für die Zufallszahl s einen beliebigen Wert verwendet. Das PASC Protokoll würde trotzdem vollständig ablaufen und beide würden die Sitzungsschlüssel K_{MAC} und K_{ENC} berechnen. Die Sitzungsschlüssel sind zwar nicht identisch, diese Tatsache bleibt dem Chip jedoch unbemerkt. Dieses Szenario erinnert an das Diffie-Hellman Problem: Beide Teilnehmer haben zwar einen gemeinsamen Schlüssel berechnet, aber sind sich nicht sicher, mit wem sie eigentlich kommunizieren, weil keine Authentifizierung statt gefunden hat. Trotz dieser Schwachstelle ist das Gefahrenpotenzial nicht offensichtlich. Auf der einen Seite sind die Daten, die der Ausweis dann vielleicht schickt, unbrauchbar, weil sie mit einem anderen Schlüssel geschützt sind. Andererseits geht der Ausweis von einem authentischen Lesegerät aus und bemerkt nicht, dass er eigentlich jede Kommunikation abbrechen sollte. Ein Lesegerät ohne Kenntnis des Schlüssels bzw. Passwortes π kann auch ein Indiz dafür sein, dass der Ausweisinhaber seine Einwilligung zur Kommunikation nicht gegeben hat oder es sich um einen Denial-of-Service Angriff handelt. Der entscheidende Faktor in Bezug auf die Sicherheit ist, wie der Ausweis nach Abschluss des PASC Protokolls fortfährt. Sendet der Ausweis weitere (verschlüsselte) Daten, könnte dies ein Sicherheitsproblem implizieren, weil ggf. Informationen preisgegeben werden, die nur für authentifizierte Lesegeräte bestimmt sind. Wartet der Chip jedoch auf den Empfang neuer Informationen, wie beispielsweise für eine Chip-Authentifizierung, dann ist das Risiko geringer einzuschätzen. Der Ausweis würde Daten, die mit einem anderen Schlüssel verschlüsselt wurden, empfangen und diese wären für ihn unverständlich. Das potenzielle Sicherheitsrisiko hängt demnach von dem weiteren Vorgehen des elektronischen Personalausweises ab. PASC verfolgt hier Ansätze von bereits etablierten und standardisierten Protokollen, setzt diese aber nicht konsequent durch.

In der Literatur zu PASC fällt des öfteren das Argument der „besseren Performance“ im Vergleich zu PACE [48]. Empirische Zahlen zur genauen Laufzeit von PASC gibt es allerdings nicht. Eine Analyse der Performance von PACE ist in [49] aufgeführt, die eine Laufzeit in dieser Implementierung von 945 ms bestimmt. Dabei fallen 677 ms (71,6 %) der Operationen auf die Chipkarte. Um einen Vergleich der Performance beider Protokolle zu betrachten, muss PASC die optionale Authentifizierung der Teilnehmer durchführen. Die Performance-Steigerung, die PASC gegenüber PACE objektiv erreichen kann, liegt in der zweiten Phase der Protokolle, in der Informationsaustausch und Berechnungen des neuen Punktes auf der elliptischen Kurve erfolgen. PASC benötigt in diesem Schritt eine APDU mehr als PACE. Laut [49] fallen 32 ms (3,3 %) der Ausführungszeit von PACE (945 ms) auf die kontaktlose Kommunikation, dass heißt, der Austausch der Informationen per APDU. Bei PACE fallen zehn Nachrichten an, was im Durchschnitt 3,2 ms pro APDU bedeutet. Dabei ist zu beachten, dass einige Nachrichten unter und andere über diesem Durchschnittswert liegen, was anhand der Länge sowie der Art der APDU zu begründen ist. PASC würde infolgedessen eine 3,2 ms höhere Ausführungszeit haben als PACE, was jedoch marginale 0,3 % zur Folge hätte, die wir vernachlässigen möchten. Entscheidend sind aus diesem Grund die unterschiedlichen Berechnungen die PASC und PACE in der zweiten Phase durchführen.

Ein essentieller Bestandteil für die Sicherheit eines kryptographischen Protokolls ist ein mathematischer Beweis. Um die Notwendigkeit solcher Beweise zu verdeutlichen wird gerne das Needham-Schroeder Protokoll [50] erwähnt. In der asymmetrischen Variante des Protokolls wurde erst 17 Jahre nach der Veröffentlichung eine Schwachstelle entdeckt [51]. Um die Sicherheit der Daten auf dem elektronischen Personalausweis zu garantieren, ist es daher unausweichlich für die verwendeten Protokolle einen solchen Beweis zu erbringen. Insbesondere weil es sich um persönliche und biometrische Daten handelt, die sehr lange Bestand haben und bei einem Missbrauch erhebliche negative Auswirkungen im finanziellen und gesellschaftlichen Bereich haben können. Solche Beweise werden anhand eines Modells bzw. mit Modell-Checker erstellt. Für PACE und PASC existieren bereits mathematische Beweise [52] [48], die mit Hilfe von Sicherheits-Modellen Bezug auf verschiedene Aspekte der Sicherheit nehmen.

4 Mobile Endgeräte und mobile Kommunikation

Das mobile Zeitalter, das wir heute als selbstverständlich erachten, hat sich rasant entwickelt. Das erste mobile Telefonieren ist auf das Jahr 1926 datiert, auf der Zugstrecke der Deutschen Reichsbahn zwischen Hamburg und Berlin [53].

> *„Das Leitbild Universal Personal Telecommunications, das heißt die Vorstellung, dass jedes Individuum von jedem Ort der Welt jedes andere Individuum erreichen kann, ist schon mindestens achtzig Jahre alt.“ — Günter Burkart* [53]

Heute verfügen Handys über einen Musikplayer, eine Kamera und auch das Internet hält Einzug in mobile Geräte. Der Trend ist offensichtlich: Jegliche Techniken, die wir von den heimischen Computern her kennen, werden auf mobilen Endgeräten etabliert. Die ersten Mobilfunkanbieter haben damit begonnen ihre Internetplattformen ihren Kunden kostenlos über das Handy zugänglich zu machen, andere werben mit einer kostenlosen Menge an Datenvolumen. Es ist also nur eine Frage der Zeit, bis mobiles Internet zum Alltag gehört und der Empfang einer E-Mail genauso selbstverständlich ist wie der einer Short Message Service (SMS). Die Technik geht jedoch schon einen Schritt weiter: Die NFC Technologie ermöglicht die Kommunikation mit RFID-Chips und eröffnet damit ein weiteres Anwendungsgebiet für mobile Geräte.

4.1 Handy Betriebssysteme

Die wesentlichen Aufgaben eines Betriebssystems besteht in der Verwaltung der Hardwarekomponenten und der Interaktion mit dem Benutzer. Die DIN 44300[1] definiert ein Betriebssystem als:

> *„Die Programme eines digitalen Rechensystems, die zusammen mit den Eigenschaften dieser Rechenanlage die Basis der möglichen Betriebsarten des digitalen Rechensystems bilden und die insbesondere die Abwicklung von Programmen steuern und überwachen.“*

Demnach bildet ein Betriebssystem die Basis eines Rechensystems zur Ressourcen-Verwaltung und zum Ausführen, Steuern und Überwachen von Programmen. Ein Betriebssystem dient so als Abstraktionsschicht für den Benutzer und Programme, um die Komplexität der einzelnen Komponenten zu verbergen. Durch die starke Verbreitung mobiler Endgeräte und den immer neuen Anwendungsmöglichkeiten hat sich die Vielfalt an Funktionalität und Anwendungen von klassischen stationären Rechnersystemen auch auf die mobile Welt ausgeweitet. Vom klassischen Telefonieren, der Adressverwaltung und dem Versand von Kurznachrichten bis hin zum Internet, Musik, Kamera u.v.m. sind heute viele Funktionen mobil verfügbar. Dadurch wachsen auch die Anforderungen an Betriebssysteme für mobile Endgeräte. Zur Einsparung von Kosten und zur einfachen Entwicklung neuer Applikationen stehen heute viele Technologien der Programmierung auch für mobile Endgeräte zur Verfügung.
In den folgenden Kapiteln 4.1.1 bis 4.1.4 betrachten wir einige der gängigen Betriebssysteme heutiger Geräte.

[1] jetzt ISO/IEC 2382 [54]

4.1.1 Android

Angeführt von der Google Inc. entwickelt die Open Handset Alliance das Betriebssystem Android[2] für Handys und Smartphones. Grundlage ist der Linux-Kernel in der Version 2.6 sowie angepasste Java- und C-Bibliotheken. Zusätzlich existieren weitere Bibliotheken für Multimediaanwendungen, Webbrowser, 3D-Anwendungen und Datenbanken. Teile des Quellcodes sind frei zugänglich und werden unter der Apache-Lizenz[3] vertrieben. Mit dem Android Software Development Kit lassen sich auf Grundlage von Java eigene Applikationen für das Betriebssystem entwickeln.

4.1.2 Microsoft Windows Mobile

Das auf Windows CE basierende Betriebssystem für PDAs, Smartphones und mobile Multimediageräte wird seit 2002 unter dem Name Windows Mobile[4] vermarktet. Die große Ähnlichkeit der Bedienung, der Programme und des Designs zur Desktop-Version lässt sich nicht abstreiten, im Detail kommt jedoch bei Windows Mobile ein anderer Kernel zum Einsatz, der es nicht gestattet, „Windows-Programme“ auf der mobilen Version auszuführen. Im Laufe der Entwicklung haben aber immer mehr bekannte Programme Einzug in die mobile Version gehalten. So findet sich heute Microsoft Office, Microsoft Outlook und auch der Windows Media Player auf dem Handy wieder.

4.1.3 Openmoko

Die Entwicklung von Openmoko[5] wird vorwiegend von der Openmoko Inc. vorangetrieben sowie einer Gemeinde an externen Softwareentwicklern. Openmoko verkörpert den Sinn der *offenen Mobilkommunikation*, das heißt die Entwicklung eines Betriebssystems für Smartphones im Sinne von freier Software. Basierend auf dem Linux-Kernel in der Version 2.6 ist der Quellcode aller Komponenten frei zugänglich. Für das erstmals im Jahre 2006 vorgestellte Projekt sind zur Zeit zwei Geräte verfügbar. Für das Neo 1973 als auch für das Neo FreeRunner sind nicht nur der Quellcode des Betriebssystems, sondern auch jegliche Spezifikationen und Schaltpläne öffentlich. Wegen zu geringen Absatzzahlen und fehlendem Budget für Support und Entwicklung sah sich der Hersteller im April 2009 jedoch genötigt das Projekt einzustellen.

4.1.4 Symbian OS

Das Betriebssystem findet sich sowohl auf Handys als auch PDAs und wird von der Symbian Ltd.[6] entwickelt. Die Ziele sind ein sicheres und benutzerfreundliches Betriebssystem. Die Entwicklung von Software erfolgt unter anderem in C++, Java und Flash Lite, darüber hinaus bietet Symbian eine Unterstützung von relationalen Datenbanken an. Als Mitglied der Symbian Foundation, einem Zusammenschluss führender Handy-Hersteller, ist der Quellcode frei zugänglich, für die nächsten Jahre ist auch eine teilweise Offenlegung als Open Source geplant. Laut dem britischen Marktforschungsinstitut Canalys [55] ist Symbian mit 46,6% der Marktführer bei Smartphone-Betriebssystemen, gefolgt von Apple mit 17,3% und RIM mit 15,2%. Den vierten Platz belegt Microsoft mit 13,6% und Linux kommt mit 5,1% auf Position fünf. Wie bei Marktführern üblich, geraten diese schnell ins Visier Krimineller, die mit der Entwicklung von Schadsoftware wie Viren, Trojaner usw. die Schutzziele der Software angreifen.

[2] http://www.android.com
[3] http://www.apache.org/licenses/
[4] http://www.microsoft.com/windowsmobile
[5] http://www.openmoko.com
[6] http://www.symbian.com

Andreas P. Heiner stellt in einem Artikel [56] die sichere Installation von zusätzlicher Software vor, um so die Sicherheit von Symbian OS zu erhöhen. Heute kann Software ohne ein gültiges Zertifikat von SymbianSigned nicht installiert werden.

4.1.5 Zusammenfassung

Die Betriebssysteme auf mobilen Endgeräten verfügen über keine Benutzerverwaltung, demnach auch über keine Authentifizierung der Benutzer. Die klassische Eingabe einer PIN beim Starten eines Mobilfunktelefons erweckt diesen Anschein, ist aber keine Authentifizierung des Benutzers, sondern vielmehr eine Authentifizierung gegenüber der Subscriber Identity Module (SIM) Karte. Mittels der SIM-Karte ist ein Mobilfunkprovider in der Lage, einen Kunden zu identifizieren und ihm Zugriff auf seine Infrastruktur zu gewähren. Einige Systeme, wie Symbian OS, bieten jedoch die Möglichkeit ein Mobilfunktelefon mit einem sogenannten Gerätecode zu versehen. Dadurch wird die Zuordnung Kunde und SIM-Karte um eine Kopplung zwischen Kunde und Gerät erweitert. Betriebssysteme auf mobilen Endgeräten bleiben jedoch Ein-Benutzer-Systeme, im Gegensatz zu verschieden Benutzerkonten bei klassischen Desktop-Betriebssystemen.
Wie Symbian setzt Windows Mobile auf Zertifikate zur Zugriffskontrolle von Applikationen. Es existieren drei Kategorien an Zugriffsrechten: *Privilegiert*, *Normal* und *Blockiert*. Applikationen laufen standardmäßig mit normalen Zugriffsrechten, d. h. sie haben nur einen beschränken Zugriff auf Funktionen und Bereiche der Datenstruktur. Microsoft bietet jedoch keine Zertifizierung eigener Entwicklungen, sondern verweist auf die Hersteller der Geräte bzw. auf die Mobilfunkprovider. Im Gegensatz zur zentralen Zertifizierung durch SymbianSigned ist es diesbezüglich schwieriger eigene Applikationen für alle Geräte anzubieten.
Eine Anforderung, die mobiles Internet in sich birgt, ist eine Authentifizierung der Kommunikationsteilnehmer und eine Verschlüsselung der Verbindung, um so Vertraulichkeit und Integrität der übertragenen Daten sicherzustellen. Als Pendant zum Transport Layer Security (TLS) Protokoll wurde Wireless Transport Layer Security (WTLS) [57] entwickelt, um den geringeren Rechen- und Speicherkapazitäten mobiler Geräte gerecht zu werden. WTLS bietet eine anonyme, eine Server- und eine Client-Authentifizierung. Die anonyme Authentifizierung dient ausschließlich dem Austausch bzw. der Berechnung eines gemeinsamen Schlüssels für eine sichere Kommunikation; Zertifikate der Kommunikationsteilnehmer werden im Gegensatz zu Server- und Client-Authentifizierung nicht benötigt. WTLS wird von allen Betriebssystemen unterstützt und kommt bei einer Kommunikation über das Wireless Application Protocol (WAP) zum Einsatz.
In dem Artikel *Security Comparison of Mobile OSes* [58] werden eine Reihe weiterer technischer Unterschiede von Betriebssystemen auf mobilen Endgeräten dargestellt. Der Autor Arto Kettula macht insbesondere deutlich, dass Sicherheit bei mobilen Geräten ein Kernpunkt ist, der bereits in der Entwicklung eine wesentliche Rolle spielen muss. Eine nachhaltige Umsetzung von Sicherheitsmechanismen durch die Hersteller bzw. durch Installation einer Software (z.B. Virenscanner) von einem Dritt-Anbieter sollte eher kritisch gesehen werden. Ein Handy ist heute mehr als nur ein schnurloses Telefon – viele nutzen es zum Beispiel als Tresor um ihre Geheimnummern für EC- und Kreditkarten zu speichern und wiegen sich mit der Annahme in Sicherheit, dass sie entscheiden können, welche Daten sie versenden.

4.2 Drahtlose Kommunikationstechniken

Die Techniken für einen drahtlosen und mobilen Informationsaustausch gehen heute über das klassische Telefonieren und Versenden bzw. Empfangen von Kurznachrichten (SMS) hinaus. Eine drahtlose Verbindung zwischen zwei mobilen Geräten, aber auch zwischen diesen und dem heimischen Computer, gehören heute zum Standard. In diesem Bereich des Personal Area Network (PAN) existieren mehrere Technologien:
Die Infrarot-Schnittstelle[7] (Infrared Data Association – IrDA) war eine der ersten Technologien, um eine direkte Verbindung zwischen zwei ggf. verschiedenen Geräten herzustellen. Ein direkter Sichtkontakt ohne Hindernisse ist die Voraussetzung und der gravierende Nachteil dieser Technik. Die Übertagungsreichweite liegt bei ca. einem Meter und mit der IrDA 1.3 Spezifikation ist eine Datenrate bis zu 16 MBit pro Sekunde möglich. Heute ist diese Technik in modernen Geräten verschwunden und wurde weitestgehend durch Bluetooth ersetzt.
Bluetooth[8] ist eine Technologie zur Funkvernetzung von Geräten über kurze Distanz und besitzt keine Einschränkung an die optische Ausrichtung der Geräte. Mit einer Reichweite von bis zu 100 Metern und einem Datendurchsatz von bis zu 480 MBit pro Sekunde ist Bluetooth deutlich leistungsfähiger. Darüber hinaus lassen sich mehrere Geräte zu einem Netzwerk zusammenschließen. Im Gegensatz zur Infrarot-Technologie ist eine Kommunikation zwischen zwei oder mehreren Geräten nicht offensichtlich, was eine Reihe von Sicherheitsproblemen mit sich bringt [59]. Technische Details, Ziele und Informationen zur Architektur sind in dem Artikel *Bluetooth: Vision, Goals, and Architecture* [60] zusammengefasst. Eine weitere, bereits weit verbreitete Technologie wird zukünftig auch auf vielen mobilen Geräten stärkere Verbreitung finden: Wireless Local Area Network (WLAN). Zusammengefasst benötigen diese Kommunikationstechniken immer zwei aktive Geräte, die jeweils eine eigene Energieversorgung haben. Weitere Informationen zu Bluetooth und IrDA mit Focus auf die Sicherheit bietet der Artikel [61] von Praveen Yalagandula.
Neben diesen Technologien im Bereich des Personal Area Network (PAN) stehen ebenfalls die Übertragungstechniken des GSM-, UTMS- und HSDPA-Standard zur Verfügung. Diese werden von der 3GPP[9] (3rd Generation Partnership Project) verwaltet und sind einer globalen Kommunikation zuzuordnen. Ausführliche Details zu UMTS offeriert der Artikel *A UMTS Network Architecture* [62].

[7] http://www.irda.org
[8] http://www.bluetooth.com
[9] http://www.3gpp.org

4.3 Near Field Communication

Near Field Communication lässt sich als Erweiterung der RFID-Technologie betrachten und gestattet eine Kommunikation zwischen mobilen Geräten und physikalischen Objekten. Der Zugang zur Informationen im mobilen Umfeld bei Technologien wie IrDA, Bluetooth oder Wireless-LAN setzen aktive Komponenten voraus. Mit passiven RFID-Chips bzw. Tags können Informationen an jedem erdenklichen Ort platziert werden. Passiv bedeutet, dass keine Energieversorgung erforderlich ist und keine komplexe Hardware verwendet werden kann, sondern nur die „reinen“ Informationen auf dem RFID-Tag gespeichert sind.
Die Idee der RFID-Technologie ist, die Daten elektronisch zu erfassen. In Warenhäusern wurden in der Vergangenheit die Preise der einzelnen Produkte überwiegend durch Etiketten ausgezeichnet und somit für Kunden und Mitarbeiter ersichtlich. Die Einführung von Barcodes ermöglichte es dem Kassensystem, die Preise bzw. den jeweiligen Artikel zu identifizieren. Der Informationsgehalt bzw. Speicherplatz eines Barcodes ist jedoch sehr begrenzt. Mit Hilfe von RFID-Tags lassen sich deutlich höhere Informationsmengen speichern. RFID-Tags könnten beispielsweise in Film- bzw. Werbeplakaten oder Produktverpackungen integriert werden [63]. Zusammenfassend bieten Barcodes und RFID-Tags eine Datenerfassung durch elektronische Systeme ähnlich der MRZ des elektronischen Personalausweises (Kapitel 2).
NFC verbindet zwei bereits etablierte Technologien und ermöglicht die Nutzung von RFID-Chips bzw. Tags im mobilen Umfeld. Bei Bluetooth, Wireless-LAN usw. ist ein teilweise komplexer Verbindungsaufbau erforderlich, nicht nur in Bezug auf die Technik, sondern auch für die Benutzer. NFC bietet eine intuitive und benutzerfreundliche Verfahrensweise zum Verbindungsaufbau durch einfaches Zusammenführen zweier Komponenten. Andere Technologien benötigen im Vergleich zu NFC eine komplexere Konfiguration, mehr Zeit zum Verbindungsaufbau und haben einen höheren Energieverbrauch.
Das erste Handy mit NFC Technologie (Nokia 6131 NFC) wurde im Jahre 2007 von dem Telekommunikationskonzern Nokia[10] veröffentlicht. Auch wenn die Prognosen der Analysten von ABI Research für den Einsatz der NFC Technologie in mobilen Geräten nach unten korrigiert werden musste [64], bieten heute mehrere namenhafte Hersteller wie Motorola, Samsung, usw. NFC-fähige Geräte an. Für das Jahr 2011 prognostiziert ABI Research einen Marktanteil von 30 %.
NFC erlaubt nur eine geringe Reichweite von ca. 10 cm und gestattet ausschließlich sogenannte Punkt-zu-Punkt Verbindungen, d. h., es können immer nur zwei Komponenten miteinander kommunizieren. Die Übertragungsraten von NFC liegen bei 106, 212 oder 424 KBits pro Sekunde. Im Vergleich zur Infrarot-Schnittstelle mit bis zu 16 MBit pro Sekunde und Bluetooth mit 2,1 MBit bzw. ab Bluetooth 3.0 mit bis zu 480 MBit pro Sekunde im Nahbereich weist NFC nur geringe Übertragungsraten auf.

NFC definiert zwei Arten von Protokollen [65]:

- **Near Field Communication Interface Protocol-1 (NFCIP-1)**
  Standardisiert in ECMA-340 [66] und ISO/IEC 18092 [67] wird ein Transport-Protokoll, eine Kollisionsbehandlung, die Datenkodierung und das Datenformat definiert sowie die Verbindungstypen, die in Abbildung 4.1 erläutert sind.

- **Near Field Communication Interface Protocol-2 (NFCIP-2)**
  NFCIP-2 stellt eine Schnittstelle zu bereits etablierten Standards zur Verfügung und ist in ECMA-352 [68] und ISO/IEC 21481 [69] definiert. Ein Verbindungsaufbau ist dadurch mit ECMA-340 [66], ISO/IEC 14443 [14] und ISO/IEC 15396 [70] konformen Komponenten möglich. Als Kollisionsbehandlung ist Carrier Sense Multiple Access (CSMA) definiert, um sicherzustellen, dass NFCIP-2 konforme Komponenten keine Kommunikation anderer Geräte stören.

[10] http://www.nokia.com

NFC-Geräte im aktiven Modus sind in der Lage Daten eines RFID-Chips zu lesen, im passiven Modus emulieren sie einen RFID-Chip und gestatten so anderen aktiven NFC-Geräten Daten zu lesen.

Einheit A		Einheit B	Beschreibung
Aktiv (Initiator)	⟵	Passiv (Ziel)	Elektromagnetisches Feld wird von Einheit A erzeugt. Die Datenübertragung findet von Einheit B nach A statt.
Aktiv (Initiator/Ziel)	⟷	Aktiv (Initiator/Ziel)	Beim Senden von Daten aktiviert die jeweilige Einheit ein elektromagnetisches Feld. Es kann immer nur ein aktives elektromagnetisches Feld geben.
Passiv (Ziel)	⟵	Aktiv (Initiator)	Elektromagnetisches Feld wird von Einheit B erzeugt. Die Datenübertragung findet von Einheit A nach B statt.

Abbildung 4.1.: NFC Verbindungstypen

Das von einem NFC-Gerät erzeugte elektromagnetische Feld ist nicht wie bei der Infrarot-Schnittstelle auf eine Richtung bzw. einen kleinen Radius begrenzt, sondern dehnt sich, wie bei einem Wireless-LAN, in alle Richtungen aus. Eine Verbindung zwischen zwei aktiven Geräten kann aufgebaut werden, wenn sich die elektromagnetischen Felder beider Kommunikationsteilnehmer überschneiden (siehe Abbildung 4.2). Im Falle eines passiven RFID-Tags ist es ausreichend, wenn sich der Tag innerhalb des Radius befindet.

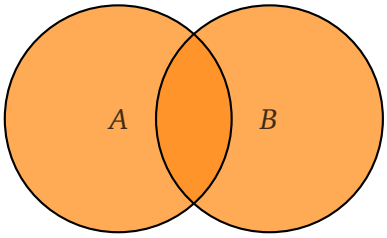

Abbildung 4.2.: NFC Verbindungsradius

4.3.1 NFC Sicherheit

Near Field Communication basiert im Wesentlichen auf der RFID-Technologie. Sicherheitsrisiken für RFID stellen demnach auch eine Bedrohung für NFC dar. Die Aritkel *RFID Systems and Security and Privacy* [71] und *RFID systems: A survey on security threats and proposed solutions* [72] illustieren eine Vielzahl der Sicherheitsrisiken und -bedrohungen der RFID-Technologie. Im folgenden Abschnitt möchten wir einige Sicherheitsrisiken diskutieren:

Abhören einer Verbindung

Die Übertragungsreichweite von NFC beträgt ca. 10 cm. Dieser Wert hängt jedoch von einer Vielzahl von Parametern ab. Um eine Kommunikation zu belauschen, muss sich ein Angreifer innerhalb der beiden Radien der elektromagnetischen Felder befinden (siehe Abbildung 4.3) bzw. innerhalb eines Radius, dann kann aber ggf. nur der Datenaustausch eines Teilnehmers erfasst werden. Betrachten wir einen Angreifer, der eine Kommunikation abhören möchte, dann ist die Qualität bzw. Leistungsfähigkeit der verwendeten Hardware, wie Antenne und Signal-Kodierer, maßgeblich für die Reichweite, in der ein Angreifer Verbindungen abhören kann. Weitere Faktoren, die die Reichweite beeinflussen können, sind physikalische Hindernisse wie Wände oder Personen. Laut [73] liegt die Reichweite für einen Angreifer eine Kommunikation zu belauschen bei aktiven NFC-Geräten bei 10 Metern und bei passiven bei 1 Meter. Um die Vertraulichkeit einer Kommunikation aufrechtzuerhalten, kann eine verschlüsselte Verbindung zwischen zwei Teilnehmern aufgebaut werden. Dabei eignen sich bereits etablierte Protokolle oder beispielsweise Secure NFC (NFC-SEC) [74].

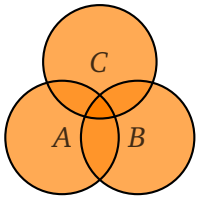

Abbildung 4.3.: NFC Sicherheit: Abhören einer Verbindung

Datenmanipulation

Unter Datenmanipulation verstehen wir das Verändern, Einfügen und Unterbinden einer Datenübertragung zwischen NFC bzw. RFID Komponenten. Diese Angriffe bedrohen die Schutzziele der Vertraulichkeit, Integrität und Verfügbarkeit. Die einfachste Möglichkeit für einen Angreifer ist das Unterbinden bzw. Stören einer Übertragung. Durch gleichzeitiges Aktivieren eines Magnetfeldes und Senden von Daten können sich die einzelnen Magnetfelder bzw. Frequenzen überlagern und die Datensignale stören. Dabei muss sich der Angreifer lediglich im Übertragungsradius eines der beiden Kommunikationsteilnehmer befinden. Dieser Angriff kann eine bestehende Verbindung korrumpieren, aber auch einen Verbindungsaufbau unmöglich machen. Standardkonforme NFC-Geräte nach NFCIP-2 prüfen auf bereits bestehende Magnetfelder, um laufende Übertragungen nicht zu stören. Diese Eigenschaft bietet jedoch keinen Schutz gegen diese Angriffe. Unsere Betrachtungen haben sich aber auf aktive NFC-Geräte beschränkt. Denkbar wären ebenfalls manipulierte RFID-Tags, die von einem NFC-Gerät ausgelesen werden. Wie in Artikel [75] beschrieben, können manipulierte RFID-Tages ein Handy zum Absturz bringen. Würde ein manipulierter RFID-Tag in der Nähe eines authentischen RFID-Tags postiert, könnte das Auslesen der Informationen verhindert werden.

Im Gegensatz zum reinen Korrumpieren einer Verbindung stellt das Verändern und das Einfügen weiterer Daten ein erheblich größeres Sicherheitsrisiko dar. Betrachten wir zuerst den Versuch eines Angreifers Daten in eine bestehende Verbindung zweier Kommunikationsteilnehmer einzufügen. Durch die kurze Übertragungsreichweite der NFC-Geräte besteht nur eine geringe Verzögerung zwischen dem Versand und dem Empfang der Daten. Diese Zeitspanne zwischen Verarbeitung empfangener Daten und dem Erzeugen einer Antwort könnte ein Angreifer nutzen, um Daten an den Absender zu schicken. Als Gegenmaßnahme könnte ein antwortendes Gerät während der Generierung der Antwort auf andere Magnetfelder prüfen, um so ggf. das Einfügen von Daten zu erkennen.
Das Verändern von übertragenden Daten stellt eine Herausforderung dar, weil Längenangaben, Prüfsummen und Sequenznummern der Daten (bzw. Pakete) berücksichtigt und ebenfalls modifiziert werden müssen. Ziel des Angreifers ist es, das Signal unbemerkt mit einem weiteren so zu überlagern, dass die gewünschten bzw. manipulierten Daten den Empfänger erreichen. Laut [73] ist diese Manipulation abhängig von der verwendeten Datenkodierung. Bei dem Einsatz der modifizierten Miller-Kodierung ist die Wahrscheinlichkeit gering, bei der Manchester Kodierung ist ein solcher Angriff umsetzbar.

Man-in-the-Middle

Ein Man-in-the-Middle-Angriff ist ein Szenario, bei dem sich eine unerwünschte und unautorisierte Person zwischen zwei Kommunikationsteilnehmern befindet und die übertragenen Informationen mithört und ggf. manipuliert bzw. unterdrückt. Wie in Abbildung 4.4 dargestellt, möchten Alice und Bob vertrauliche Informationen austauschen. Sie befinden sich aber unwissentlich in einer Konversation mit einer 3. Partei.
Eve fungiert als Angreifer und ist in der Lage, jegliche Information zu beobachten und zu manipulieren. Das Ziel von Eve ist es, die Kontrolle über die Kommunikation zu erlangen und Alice und Bob im Glauben zu lassen, dass sie unbeeinflusst kommunizieren. Eve gibt sich bei der Kommunikation mit Alice als Bob aus und gegenüber Bob als Alice.

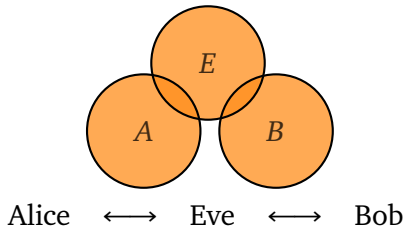

Abbildung 4.4.: NFC Sicherheit: Man-in-the-Middle-Angriff

Für Alice und Bob ist die Anwesendheit von Eve nicht ersichtlich. Um eine verschlüsselte Verbindung aufzubauen, benutzen Alice und Bob zum Beispiel das Diffie-Hellman-Protokoll. Durch den Austausch von Informationen sind Alice und Bob in der Lage, einen gemeinsamen Schlüssel zu berechnen. Dieser Schlüssel dient dann zum Verschlüsseln der zu übertragenden Nachrichten. Im Falle eines Man-in-the-Middle-Angriffs vereinbaren jedoch Alice und Eve sowie Eve und Bob einen gemeinsam Schlüssel. Alice und Bob verbleiben in der Annahme, sicher zu kommunizieren, jedoch erlangt Eve die vollständige Kontrolle über die Verbindung. Intuitiv können wir annehmen, dass ein solches Angriff-Szenario bei NFC nicht möglich ist, weil die Reichweite von 10 cm zu gering erscheint, dass ein Angreifer innerhalb dieses Radius ein weiteres Gerät zum Abhören platzieren könnte. Die Spezifikation von NFC definiert eine Prüfung auf bereits in Reichweite befindliche elektromagnetische Felder, bevor ein Gerät ein eigenes elektromagnetisches Feld zur Kommunikation aufbauen darf. In dem Artikel [73] werden die drei vorgestellten Varianten einer NFC Verbindung betrachtet (siehe Abbildung 4.1) und auf die Möglichkeit eines

Man-in-the-Middle-Angriffs analysiert. Zusammenfassend kommen die Autoren zu dem Ergebnis, dass bei Geräten, die sich konform zur Spezifikation verhalten, ein Man-in-the-Middle Angriff praktisch nicht möglich ist. Der Fokus der Analyse beschränkt sich jedoch ausschließlich auf die NFC Technologie und abstrahiert von den Softwarekomponenten, die auf den jeweiligen Geräten zum Einsatz kommen. Das eine Betrachtung der Software aber durchaus von Bedeutung ist, zeigt der Artikel *Vulnerability Analysis and Attacks on NFC-enabled Mobile Phones* [75] von Collin Mulliner. Das wesentliche Problem stellt dabei das Betriebssystem und seine einzelnen Komponenten dar. Durch manipulierte RFID-Tags ist es möglich, die Darstellung der ausgelesenen Daten so zu verändern, dass ein Man-in-the-Middle Angriff möglich wird. Abschließend lässt sich feststellen, dass die NFC Technologie durch ihre Übertragungstechnik nicht betroffen ist, jedoch die verwendete Software auf den Geräten selbst.
Die Gefahr eines Man-in-the-Middle Angriff ist nicht nur bei NFC zu analysieren, bei vielen Kommunikationstechnologien bzw. Protokollen besteht dieses Sicherheitsrisiko. Die Problematik ist beispielsweise bei Bluetooth [76] und UMTS [77] zu finden.

Zusammenfassung

Der wesentliche Aspekt bei der Betrachtung der Sicherheit und Zuverlässigkeit liegt in der Differenzierung zwischen der reinen NFC Technologie, spezifiziert durch NFCIP-1 und NFCIP-2, und der auf den mobilen Endgeräten verwendeten Software-Komponenten wie Betriebssystem, Browser, E-Mail Programm usw.. Wie der Artikel *Security in Near Field Communication* [73] illustriert, können einige Angriffe bereits durch die physikalischen Eigenschaften der NFC Technologie neutralisiert werden. Dazu zählen die Charakteristiken eines elektromagnetischen Feldes, der Datenkodierung und das Erkennen von Kollisionen durch andere Magnetfelder. Die einzelnen Software-Komponenten stellen jedoch ein erhebliches Sicherheitsrisiko dar [75], bei denen mit geringem Aufwand manipulierte RFID-Tags bereits zum Absturz führen können. Des Weiteren sind klassische Spoofing- und Phishing-Angriffe einfach umzusetzen, und so können ahnungslose Benutzer auf teure und kostenpflichtige Webseiten oder Rufnummern umzuleitet oder Daten auszuspäht werden.

5 Prototyp

5.1 Entwicklungsumgebung

Um einen weitestgehend plattform-übergreifenden bzw. unabhängigen Prototypen zu entwickeln, fiel die Entscheidung auf die Programmiersprache Java. Als integrierte Entwicklungsumgebung (IDE) kam die Software Eclipse zum Einsatz sowie das Plugin Mobile Tools for Java. Als Basisversion diente die Java SDK 6 mit den Erweiterungen Java Micro Edition SDK und Sun Java Wireless Toolkit for CLDC. Darüber hinaus fand das Nokia Series 40 Nokia 6212 NFC SDK Verwendung, das einen Emulator für das Nokia 6212 classic bietet und die weiteren benötigten Java-Spezifikationen (JSR) wie die JSR 257 Contactless Communication API [78] beinhaltet. Für die kryptographischen Verfahren und Algorithmen sowie das Kodieren bzw. Dekodieren der ASN.1 (Abstract Syntax Notation One) Datenstrukturen wurde der Cryptography Service Provider (CSP) Bouncy Castle verwendet. Die verwendeten Softwareprodukte sind in Abbildung 5.1 verzeichnet.

Software	Version
Eclipse	3.5.0 Galileo
Mobile Tools for Java	1.0.0
Java SE Development Kit JDK	6 Update 11
Java Micro Edition SDK	3.0
Nokia Series 40 Nokia 6212 NFC SDK[1]	1.0

Abbildung 5.1.: Entwicklungsumgebung

5.2 Hardwarekomponenten

Der Telekommunikationskonzern Nokia bietet zur Zeit zwei NFC-fähige Geräte an: Das Nokia 6131 NFC und das Nokia 6216 classic. Alle Geräte basieren auf der *Series 40* Benutzeroberfläche. Für die Entwicklung des Prototyps kam das Nokia 6216 classic[2] zum Einsatz. Neben zahlreichen Multimediafunktionen verfügt das Gerät über eine Bluetooth-, USB- und eine NFC-Schnittstelle. Bei der Implementierung wurde auch der Chipkartenleser SDI010 von SCM Microsystems verwendet.

[1] http://www.forum.nokia.com/info/sw.nokia.com/id/5bcaee40-d2b2-4595-b5b5-4833d6a4cda1/S40_Nokia_6212_NFC_SDK.html

[2] http://www.forum.nokia.com/devices/6212_classic

5.3 Implementierung

Nachdem wir in Kapitel 3.3.2 das PACE Protokoll und in Kapitel 4.3 die NFC Technologie als Schnittstelle zwischen einem Handy und dem elektronischen Personalausweis ausführlich erläutert haben, stellen wir jetzt die technische Seite einer Implementierung des PACE Protokolls auf einem NFC-fähigen Mobilfunktelefon vor. Neben der reinen PACE Implementierung wurden weitere Komponenten entwickelt, die insbesondere für die Weiterentwicklung des Prototyps von Bedeutung sind, aber auch durch die eingeschränkte Funktionalität der Java Micro Edition notwendig waren. Die Ziele bei der Entwicklung waren ein robustes Design im Sinne von weitestgehend unabhängigen Komponenten, eine klare und verständliche Struktur sowie eine große Flexibilität in Hinblick auf Weiterentwicklung und Veränderungen. Im Wesentlichen galt es eine Architektur nach dem *Model View Controller* Prinzip zu entwickeln, d. h. eine klare Trennung zwischen Daten, deren Darstellung und der Steuerung zu erreichen. Zusätzlich wurden verschiedene Design Patterns (Entwurfsmuster) verwendt, um den Qualitätsstandards des Software Engineering gerecht zu werden. Abbildung 5.3 illustriert die einzelnen Komponenten des entwickelten Prototyps.

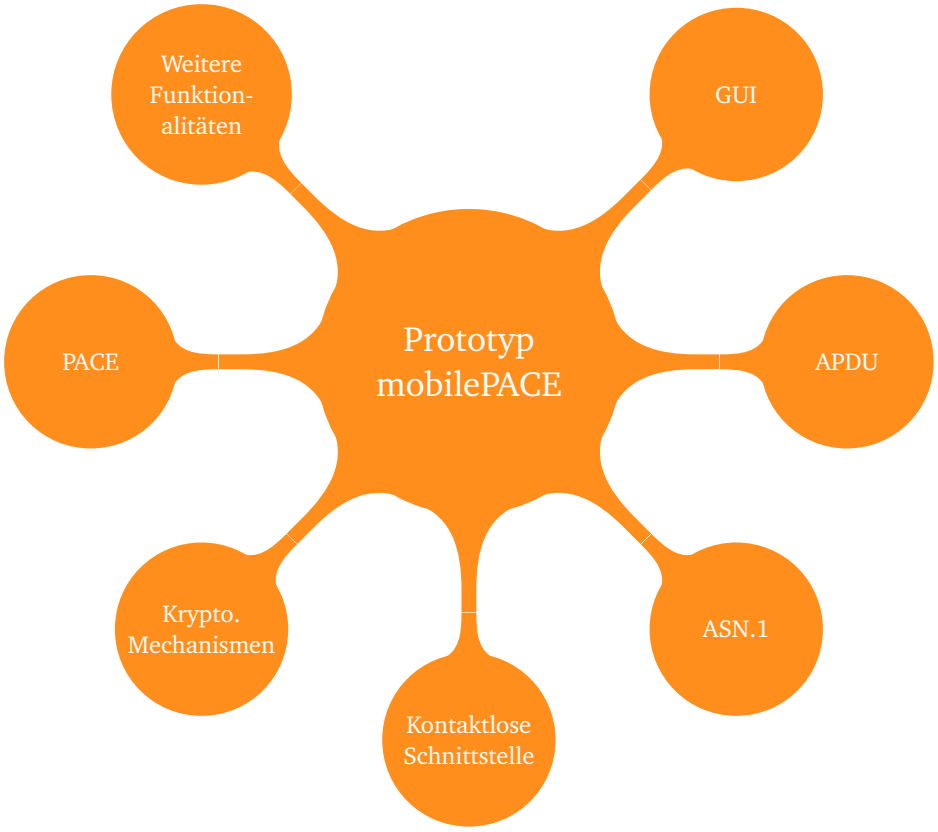

Abbildung 5.3.: Komponenten

Paketübersicht

Zu Beginn möchten wir eine Übersicht der Pakete geben und deren Aufgaben bzw. die Funktionalität der darin enthaltenen Klassen beschreiben. Details der wichtigsten Klassen werden in den Kapiteln 5.3.1 bis 5.3.7 vorgestellt.

- `de.tud.cdc.mecca.apdu`
  Das Paket beinhaltet mehrere Schnittstellen für Application Protocol Data Units (APDU), sowie Implementierungen der benötigten APDUs.
- `de.tud.cdc.mecca.asn1`
  Klassen zum Extrahieren und Verarbeiten von Daten im ASN.1 Format sind in diesem Paket enthalten.
- `de.tud.cdc.mecca.card`
  Stellt eine Schnittstelle für die kontaktlose Kommunikation bereit und eine konkrete Implementierungen für eine Übertragung mittels NFC.
- `de.tud.cdc.mecca.common`
  Enthält Klassen die eine *allgemeine* Funktionalität bereitstellen.
- `de.tud.cdc.mecca.crypto`
  Implementierungen der kryptographischen Komponenten für das PACE Protokoll.
- `de.tud.cdc.mecca.eac.protocols`
  Beinhaltet die Implementierung des PACE Protokolls.
- `de.tud.cdc.mecca.gui`
  Das Paket stellt die Klassen für die graphische Benutzeroberfläche bereit.
- `de.tud.cdc.mecca.iso`
  Implementierungen der benötigten Definitionen der ISO/IEC 7816 [9] und EAC 2.01 Spezifikation [3].
- `de.tud.cdc.mecca.main`
  Hauptprogramm des Prototyps.
- `de.tud.cdc.mecca.observer`
  Implementierung des Observer Pattern [79].

Die Klassendiagramme in den Kapiteln 5.3.1 bis 5.3.7 verwenden die Unified Modeling Language (UML). Details zur Notation sind unter anderem dem Buch *The Unified Modeling Language Reference Manual* [80] zu entnehmen. Aus Gründen der Übersicht wurde in einigen Diagrammen auf Details, darunter *private* Methoden, Abhängigkeiten und Verbindungen der einzelnen Klassen, verzichtet.

5.3.1 Graphische Benutzeroberfläche

Der Prototyp verfügt über eine graphische Benutzeroberfläche (GUI - Graphical User Interface). Im Hauptmenü stehen die Punkte Verbinden, Hilfe und Beenden zur Verfügung. Nach der Auswahl Verbinden kann zwischen der Verwendung einer PIN oder CAN als Passwort zur Authentifizierung gewählt werden. Im nächsten Schritt stellt die GUI eine Eingabemaske für das Passwort bereit. Die Eingabe wird validiert, d. h. die Anzahl der Zeichen des Passwortes wird überprüft. Wird kein Passwort eingegeben oder besteht das eingegebene Passwort aus weniger als sechs Ziffern, erfolgt eine Fehlermeldung und die Eingabe kann wiederholt werden. Nach der Eingabe des Passwortes wird das PACE Protokoll initialisiert und dessen Status in Form einer Fortschrittsleiste und kurzen Meldungen auf der GUI angezeigt. Die graphische Benutzeroberfläche besteht im Wesentlichen aus folgenden Klassen:

Paket: `de.tud.cdc.mecca.gui`

- `de.tud.cdc.mecca.gui.GUI`
  Die Klasse implementiert die einzelnen Navigationsmenüs und Eingabemasken.
- `de.tud.cdc.mecca.gui.EACGUI`
  Das Anzeigen der Fortschrittsleiste und der Statusmeldung des PACE Protokolls ist in dieser Klasse realisiert.

Die einzelnen Navigationsmenüs und Eingabemasken wurden in Methoden implementiert, um diese wiederverwenden zu können. Die Eingabemaske für das Passwort musste infolgedessen nur einmal implementiert werden und unterscheidet anhand eines Parameters, ob eine PIN oder CAN verwendet wird. Da das PACE Protokoll zusätzlich noch eine Authentifizierung per MRZ unterstützt, lässt sich die Eingabemaske auch für diesen Passworttyp sowie für den PUK verwenden. Auch die Methode für das Validieren des Passwortes ist flexibel implementiert worden. Änderungen an der Ziffernzahl bzw. Struktur der Passwörter könnten deswegen leicht umgesetzt werden, insbesondere da bis zur Einführung des elektronischen Personalausweises ggf. mit Änderungen zu rechnen ist. Auch wenn davon auszugehen ist, dass die PIN, CAN oder der PUK ausschließlich aus Ziffern besteht, ist die Methode zum Validieren der Passwörter in der Lage auch Kombinationen aus Zahlen und Zeichen zu überprüfen. Ausschnitte der graphischen Oberfläche sind in Abbildungen 5.4 aufgeführt.

Abbildung 5.4.: Graphische Benutzeroberfläche

In Hinblick auf die Erweiterung des Prototypen wurde die Klasse `EACGUI` protokoll-unabhängig implementiert. Das heißt, die Klasse stellt ausschließlich die Funktionalität des Anzeigen von Statusmeldungen und einer dazugehörigen Fortschrittsleiste bereit. Welche Meldungen angezeigt werden, ist davon unabhängig. Zu Beachten ist jedoch, dass die Fortschrittsleiste, in Hinblick auf die Prozentangabe, nur korrekte Werte liefert bzw. anzeigt, wenn definiert wurde, wie viele Statusmeldungen zu erwarten sind.

5.3.2 Application Protocol Data Units

Eine Application Protocol Data Unit (APDU) nach dem ISO/IEC 7816 Standard [9] ist eine Dateneinheit bzw. ein Container für Befehle und Daten, mit deren Hilfe ein Datenaustausch zwischen Chip und Lesegerät realisiert wird. Dabei wird zwischen einer Command APDU für die Übermittlung von Kommandos und Daten an die Chipkarte und einer Response APDU als Rückmeldung auf eine Command APDU differenziert.

Der Header der Command APDU besteht aus den Feldern Class (CLA), Instruction (INS), Parameter 1 (P1) und Parameter 2 (P2), die jeweils eine Größe von einem Byte aufweisen (siehe Abbildung 5.5). Der Body setzt sich aus den Feldern Length Command (Lc), Data und Length Expected (Le) zusammen. Das Lc-Feld beschreibt die Größe bzw. Länge des Data-Feldes, wohingegen Le die erwartete Länge des Datenfeldes in der Response APDU beschreibt.

Header				**Body**		
CLA	INS	P1	P1	Lc	Data	Le
4 Bytes				0-3 Bytes	Lc Bytes	0-3 Bytes

Abbildung 5.5.: Command APDU [9]

Die Response APDU besteht aus einem Feld Data und einem Feld Statuswort (SW1, SW2). Das Data-Feld wird für Antwortdaten und das SW-Feld für Statuscodes verwendet (siehe Abbildung 5.6). Bei einer Command APDU bzw. Response APDU müssen nicht alle Felder belegt bzw. verwendet werden, wenn beispielsweise auf eine Command APDU keine Antwortdaten, sondern nur eine Bestätigung erwartet wird. Weitere Details sind der ISO/IEC 7816-4 [9] zu entnehmen.

Body	**Trailer**	
Data	SW1	SW2
Le Bytes	2 Bytes	

Abbildung 5.6.: Response APDU [9]

Für die Implementierung von PACE kommen die Kommandos MSE:Set AT und General Authenticate zum Einsatz [3]. Das MSE:Set AT (MSE = Manage Security Environment, AT = Authentication Template) Kommando (Abbildung 5.7) dient der Initialisierung und Auswahl der Authentifizierungs-Methode und weist den elektronischen Personalausweis an, beispielsweise mit dem Parameter 0xC1A4, das PACE Protokoll als Authentifizierung zu verwenden. Das Data-Feld enthält weitere Informationen, wie beispielsweise den verwendeten Passworttyp (siehe Kapitel 2.2 und [3, Kapitel B.11.1]).

Header				**Body**		
CLA	INS	P1	P1	Lc	Data	Le
0x00	0x22	0xC1A4				

Abbildung 5.7.: MSE:Set AT APDU [3]

Das Kommando General Authenticate (Abbildung 5.8) kommt bei dem Austausch der protokollspezifischen Datenobjekte zwischen dem elektronischen Personalausweis und dem Lesegerät zum Einsatz [3, Kapitel B.1], d. h. bei dem Austausch der verschlüsselten Zufallszahl, den Informationen für den neuen Punkt auf der elliptischen Kurve, der flüchtigen Schlüssel PK_{PICC} und PK_{PCD} sowie der Authentifizierungstoken T_{PICC} und T_{PCD}.

Header				**Body**		
CLA	INS	P1	P1	Lc	Data	Le
	0x86	0x0000				

Abbildung 5.8.: General Authenticate APDU [3]

Die APDUs wurden in einer eigenen Klassen-Hierarchie implementiert, um die Implementierung des PACE Protokolls unabhängig von deren konkreten Struktur und Umsetzung zu halten. Der Vorteil ist, dass die Implementierung des PACE Protokolls keine Kenntnis über den Aufbau und die Statuswörter der APDUs hat, die nach ISO/IEC 7816-4 [9] und vom BSI [3, Kapitel B.1] definiert sind, sondern nur mit Objekten arbeitet. Dadurch können beispielsweise die Statuswörter zentral definiert, Änderungen leicht umgesetzt und die Implementierung der APDUs für weitere Protokolle verwendet werden.

Paket: `de.tud.cdc.mecca.apdu.interfaces`

- `de.tud.cdc.mecca.apdu.interfaces.IAPDU`
  Das Interface bildet eine Schnittstelle für alle APDUs. Die Methode `getData()` ermöglicht den Zugriff auf das Datenfeld einer Command bzw. Response APDU. Die Mehtode `toByteArray()` verknüpft die einzelnen Felder (CLA,INS, usw.) zu einem Byte-Array.

- `de.tud.cdc.mecca.apdu.interfaces.ICommand`
  Definiert eine Schnittstelle für die Command APDUs und stellt mit der Methode `getHeader()` den Zugriff auf den Header (INS, CLA, P1, P2) zur Verfügung.

- `de.tud.cdc.mecca.apdu.interfaces.IResponse`
  Definiert eine Schnittstelle für die Response APDUs. Analog zur Command APDU existiert die Methode `getTrailer()`, zusätzlich kann mittels `getStatusWord()` auf den Statuscode einer Response APDU zugegriffen werden.

Paket: `de.tud.cdc.mecca.apdu.common`

- `de.tud.cdc.mecca.apdu.common.AbstractCommandAPDU`
  Die abstrakte Klasse implementiert eine Grundfunktionalität für Command APDUs, wie der Zugriff auf das Daten- und Header-Feld einer Command APDU.
  Alle Klassen in dem Paket `de.tud.cdc.mecca.apdu.command` erweitern diese abstrakte Klasse.

- `de.tud.cdc.mecca.apdu.common.AbstractResponseAPDU`
  Die abstrakte Klasse implementiert eine Grundfunktionalität für Response APDUs, wie der Zugriff auf das Daten- und Trailer-Feld einer Response APDU.
  Alle Klassen in dem Paket `de.tud.cdc.mecca.apdu.response` erweitern diese abstrakte Klasse.

Paket: de.tud.cdc.mecca.apdu.command

- de.tud.cdc.mecca.apdu.command.CommandReadBinary
  Die Klasse implementiert den READ BINARY Befehl zum Auslesen von Datenfeldern nach ISO/IEC 7816 [9].
- de.tud.cdc.mecca.apdu.command.CommandSelectFile
  Die Klasse implementiert das SELECT FILE Kommando zum Selektieren einer Anwendung oder eines Datenbereiches nach ISO/IEC 7816 [9]. Die Klasse kann dabei sowohl mit einem File-Identifier, als auch mit einem Short-File-Identifer instantiiert werden.
- de.tud.cdc.mecca.apdu.command.GeneralAuthenticate
  Die Klasse implementiert das GENERAL AUTHENTICATE Kommando [3, Kapitel B.11.2.].
- de.tud.cdc.mecca.apdu.command.MSESetAT
  Die Klasse implementiert das MSE:SET AT Kommando [3, Kapitel B.11.1.].

Paket: de.tud.cdc.mecca.apdu.response

- de.tud.cdc.mecca.apdu.response.ResponseError
  Diese Klasse implementiert eine Response APDU, die einen Fehlercode im Statusfeld (SW1, SW2) aufweist.
- de.tud.cdc.mecca.apdu.response.ResponseNormalOperation
  Diese Klasse implementiert eine Response APDU, die im Statusfeld (SW1, SW2) eine 0x9000 aufweist, d. h., dass das Kommando erfolgreich ausgeführt wurde und nur die Informationen im Datenfeld für die weitere Bearbeitung von Bedeutung sind.

Paket: de.tud.cdc.mecca.iso.ISO7816

- de.tud.cdc.mecca.iso.ISO7816.ISO7816
  Die Klasse implementiert die Statuscodes bzw. Returncodes der ISO/IEC 7816-4 [9, Kapitel 5].
- de.tud.cdc.mecca.iso.ISO7816.ISO7816EAC
  Die Klasse implementiert die Statuscodes bzw. Returncodes der EAC 2.01 Spezifikation [3, Kapitel B.11].

Durch das Definieren von Schnittstellen (Interfaces) wurde eine einheitliche Struktur entwickelt und mit dem Auslagern von gleicher Funktionalität in abstrakte Klassen, wie der Zugriff auf den Datenteil einer APDU, mussten diese Methoden nicht mehrmals implementiert werden. Die jeweiligen Command APDUs definieren dabei nur ihre spezifischen Eigenschaften wie beispielsweise den Header.
Das gewählte Design zeigt insbesondere seine Stärke bei dem Umgang mit einer Response APDU. Bei mehr als 50 verschiedenen Statuscodes müsste innerhalb der PACE Implementierung auf eine Vielzahl dieser bei etwaigen Fehlern reagiert werden. Die Schnittstelle für die Kommunikation zwischen PACE und dem elektronischen Personalausweis, die wir noch erörtern werden, abstrahiert dabei von dieser notwendigen Fehlerbehandlung. Beim Auftreten eines Fehlercodes im Statusfeld einer Response APDU gibt die Schnittstelle eine Instanz der Klasse `ResponseError` zurück, andernfalls eine Instanz der Klasse `ResponseNormalOperation`, falls das Kommando der vorherigen Command APDU erfolgreich ausgeführt wurde. Die PACE Implementierung muss dadurch nur prüfen, um welche Instanz es sich handelt und ggf. bei einer `ResponseError` auf spezifische Fehler reagieren. Dieser Fall tritt beispielsweise auf, wenn nach einem MSE:SET AT Kommando der elektronische Personalausweis mit der Warnung antwortet, dass das Passwort deaktiviert ist. Die Klasse `ResponseError` bietet darüber hinaus die Möglichkeit, neben dem Fehlercode auch eine Fehlerbeschreibung auszugeben, was insbesondere für eine Rückmeldung bei einer graphischen Oberfläche wichtig ist. Des Weiteren muss innerhalb der PACE Implementierung nicht mit den Statuscodes in Form von hexadezimalen Werten (z.B. 0x9000) gearbeitet werden. Die Implementierung ist dadurch übersichtlicher, strukturierter und weniger fehleranfällig bei Änderungen an den APDUs und der Statuscodes.

Die Implementierung ist an die Funktionalität einer JavaCard angelehnt. Dabei wurde jedoch eine Abstraktionsschicht hinzugefügt, die nicht nur zwischen einer klassischen Command bzw. Response APDU differenziert, sondern auch den spezifischen APDUs MSE:Set AT und General Authenticate gerecht wird. Durch die Implementierung der spezifischen APDU muss innerhalb der Klasse `PACE` auch nicht deren genaue Struktur definiert werden, sondern nur die Daten übergeben werden. Der Vorteil ist, dass innerhalb der Klasse `PACE` keine hexadezimalen Werte angegeben werden müssen und die konkrete Struktur und Aufbau verborgen bleibt. Dadurch ist die Klasse `PACE` auch unabhängig von Änderungen an den APDUs. Ebenfalls müssen auch keine Längenangaben der APDUs übergeben werden, sondern nur die Daten. Die benötigte Struktur und Berechnung der Lc Felder übernehmen die Klassen der APDUs. Das Klassendiagramm in Abbildung 5.9 illustriert die Beziehungen der wesentlichen Klassen im Paket `de.tud.cdc.mecca.apdu`.

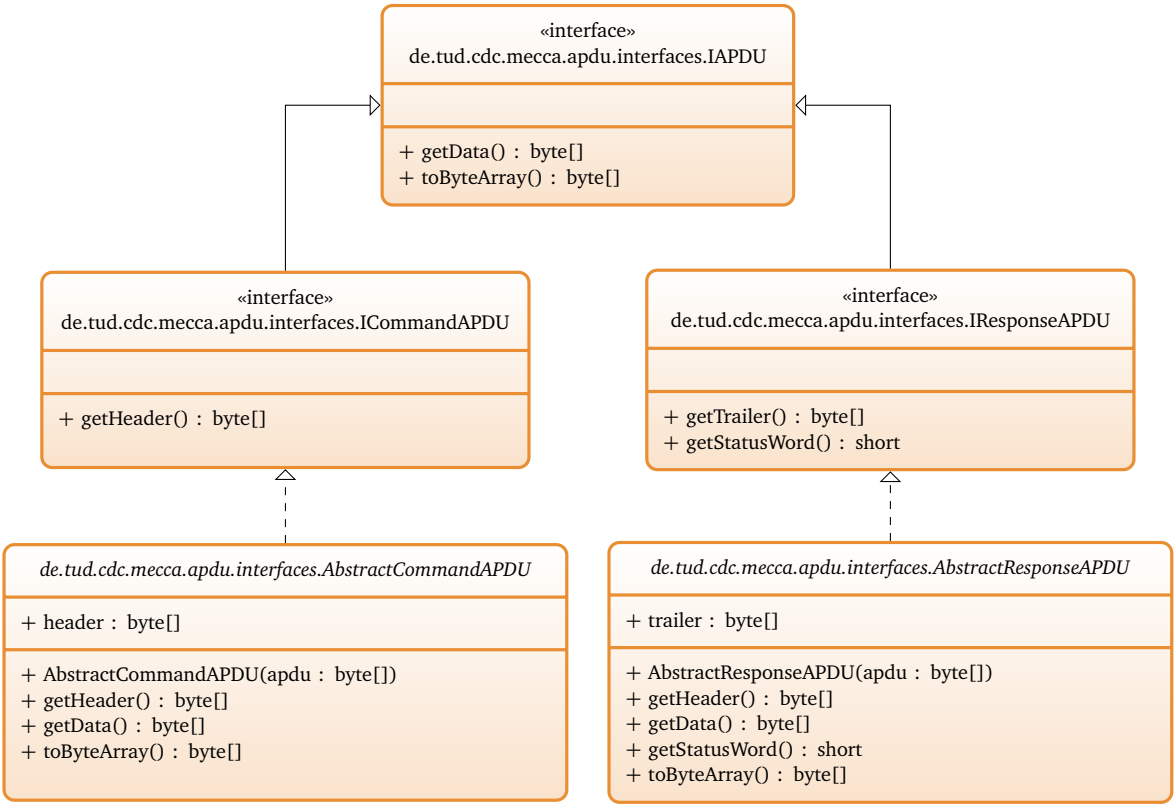

Abbildung 5.9.: Klassendiagramm: APDU

5.3.3 ASN.1

Abstract Syntax Notation One (ASN.1) ist eine Beschreibungssprache zur Definition von Datenstrukturen. Die Domain-Parameter und die verwendeten bzw. unterstützten kryptographischen Protokolle sind auf dem elektronischen Personalausweis in der Datei `EF.CardAccess` abgelegt und in ASN.1 kodiert. Für die Ausführung des PACE Protokolls muss diese Datei ausgelesen werden und die benötigten Informationen extrahiert werden.

Paket: `de.tud.cdc.mecca.asn1`

- `de.tud.cdc.mecca.asn1.ECDHAlgorithmIdentifier`
  Der Algorithm Identifier bzw. die Domain-Parameter für elliptische Kurven sind in dieser Klasse implementiert. Dabei werden die jeweiligen Informationen bzw. Werte aus der Datei `EF.CardAccess` extrahiert.

- `de.tud.cdc.mecca.asn1.TLV`
  Die Daten im `EF.CardAccess` sind im TLV-Format (Type, Length, Value) gespeichert. Die Klasse implementiert das Kodieren und Dekodieren des TLV-Formats.

Paket: `de.tud.cdc.mecca.asn1.eac`

- `de.tud.cdc.mecca.asn1.eac.ISecurityInfo`
  Die Klasse definiert eine Schnittstelle für alle SecurityInfos.

- `de.tud.cdc.mecca.asn1.eac.SecurityInfo`
  Diese Klasse bildet eine einzelne SecurityInfo ab, bestehend aus dem Object Identifier des Protokolls und den dazugehörigen benötigten bzw. optionalen Daten [3, Kapitel A.1].

```
SecurityInfo ::= SEQUENCE {
    protocol     OBJECT IDENTIFIER,
    requiredData ANY DEFINED BY protocol,
    optionalData ANY DEFINED BY protocol OPTIONAL
}
```

- `de.tud.cdc.mecca.asn1.eac.SecurityInfos`
  Die Klasse repräsentiert eine Menge der einzelnen SecurityInfos und vereint alle in der `EF.CardAccess` vorhandenen SecurityInfos. Die Klasse implementiert folgende ASN.1 Datenstruktur [3, Kapitel A.1].

```
SecurityInfos ::= SET OF SecurityInfo
```

Paket: `de.tud.cdc.mecca.asn1.eac.pace`

- `de.tud.cdc.mecca.asn1.eac.pace.PACEDomainParameterInfo`
  Die Klasse implementiert folgende ASN.1 Datenstruktur:

```
PACEDomainParameterInfo ::= SEQUENCE {
    protocol        OBJECT IDENTIFIER(id-PACE-DH | id-PACE-ECDH),
    domainParameter AlgorithmIdentifier,
    parameterId     INTEGER OPTIONAL
}
```

  Zum Zugriff auf die Daten stehen die Methoden *getProtocol()*, *getDomainParameter()* und *getParameterId()* zur Verfügung. Mit Hilfe der Methode *isPACEObjectIdentifer(DERObjectIdentifier o)* lässt sich überprüfen, ob der gegebene Object Identifer ein PACE Object Identifer der Form id-PACE-DH oder id-PACE-ECDH ist.

- de.tud.cdc.mecca.asn1.eac.pace.PACEInfo
  Die Klasse implementiert folgende ASN.1 Datenstruktur:

```
PACEInfo ::= SEQUENCE {
    protocol    OBJECT IDENTIFIER(
                id-PACE-DH-3DES-CBC-CBC |
                id-PACE-DH-AES-CBC-CMAC-128 |
                id-PACE-DH-AES-CBC-CMAC-192 |
                id-PACE-DH-AES-CBC-CMAC-256 |
                id-PACE-ECDH-3DES-CBC-CBC |
                id-PACE-ECDH-AES-CBC-CMAC-128 |
                id-PACE-ECDH-AES-CBC-CMAC-192 |
                id-PACE-ECDH-AES-CBC-CMAC-256 |
                id-PACE-DH-IM-3DES-CBC-CBC |
                id-PACE-DH-IM-AES-CBC-CMAC-128 |
                id-PACE-DH-IM-AES-CBC-CMAC-192 |
                id-PACE-DH-IM-AES-CBC-CMAC-256 |
                id-PACE-ECDH-IM-3DES-CBC-CBC |
                id-PACE-ECDH-IM-AES-CBC-CMAC-128 |
                id-PACE-ECDH-IM-AES-CBC-CMAC-192 |
                id-PACE-ECDH-IM-AES-CBC-CMAC-256),
    version     INTEGER, -- MUST be 1
    parameterId INTEGER OPTIONAL
}
```

  Die Methoden *getIdentifier()*, *getVersion()* und *getParameterId()* ermöglichen den Zugriff auf die jeweiligen Daten. Ebenfalls bietet diese Klasse die Methode *isPACEObjectIdentifer(DERObjectIdentifier o)*, die einen booleschen Wert liefert, falls es sich um einen PACE Object Identifer der Form id-PACE-DH-3DES-CBC-CBC, id-PACE-DH-AES-CBC-CMAC-128 usw. handelt.

- de.tud.cdc.mecca.asn1.eac.pace.PACESecurityInfos
  Die einzelnen SecurityInfos in der Datei EF.CardAccess sind protokoll-spezifisch. Für das PACE Protokoll sind die benötigten Informationen in der PACESecurityInfos abgelegt. Die Klasse vereinigt die Informationen aus den Klassen PACEInfo und PACEDomainParameter und bietet darüber hinaus noch weitere Funktionalitäten. Für die Initialisierung des PACE Protokolls mittels des MSE:SET AT Kommando (siehe 5.3.2) muss der verwendete Passworttyp angegeben werden (siehe 2.2 und [3, Kapitel B.11.1]). Die dazugehörigen hexadezimalen Werte sind durch die Konstanten PACE_PASSWORD_TYPE_MRZ usw. definiert. Bei der Instantiierung der Klasse PACESecurityInfos kann das verwendete Passwort angegeben werden. Standardmäßig wird die CAN als Passwort verwendet, um ggf. fehlerhafte Authentifizierungsversuche zu vermeiden.

Im Wesentlichen benötigen die Klassen CipherSuite und KDF aus dem Packet de.tud.cdc.mecca.crypto die Informationen aus der PACESecurityInfo. Dazu gehören unter anderem der verwendete Verschlüsselungsalgorithmus (AES oder Triple-DES), der Hash-Algorithmus und dessen Hash-Länge, d. h. die Anzahl der Bits des Hash-Werts. Damit diese Klassen jedoch keine ASN.1 Funktionalität benötigen und somit unabhängig vom verwendeten ASN.1 Parser sind, bietet die Klasse PACESecurityInfos die Methoden getCipherAlgorithm(), getHashAlgorithm() und getHashLength(). In Anbetracht, dass der Klasse PACESecurityInfos alle Informationen über die verwendeten kryptographischen Verfahren zur Verfügen stehen, wurde das Extrahieren und die Auswahl der verwendeten Algorithmen auch dort implementiert. Damit geht dessen Funktionalität über die eigentliche „Definition“ eines schlichten Zusammensetzen von PACEInfo und PACEDomainParameter hinaus. Aus dem bereits beschriebenen Grund die Klassen CipherSuite und KDF autonom in Hinblick auf ASN.1 zu lassen, wurde die Funktionalität erweitert.

Das Klassendiagramm in Abbildung 5.10 verdeutlicht die Beziehungen der wesentlichen Klassen im Paket `de.tud.cdc.mecca.asn1`.

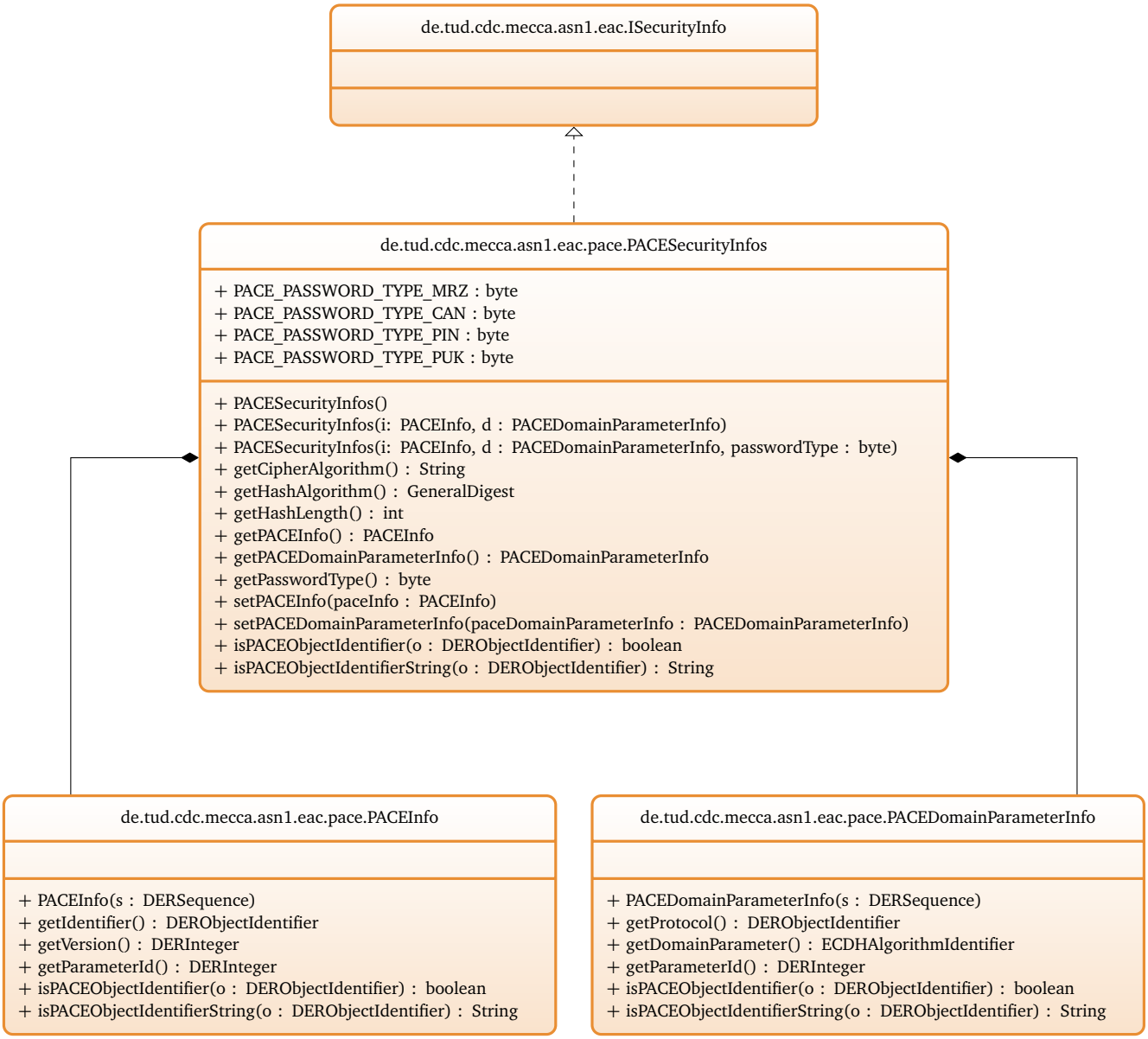

Abbildung 5.10.: Klassendiagramm: PACESecurityInfos

5.3.4 Kontaktlose Schnittstelle

Bei der Entwicklung kam neben dem Handy auch ein stationärer Kartenleser zum Einsatz. Um zwischen den einzelnen Kommunikationsschnittstellen komfortabel zu wechseln und PACE unabhängig von der verwendeten Schnittstelle zu implementieren, wurde von der konkreten Verbindung abstrahiert. Das Interface `IISO14443` definiert eine Schnittstelle, die von allen Klassen implementiert werden muss, die eine Kommunikation zum elektronischen Personalausweis herstellen können. Für die Klasse `PACE` und `CardProxy`, die eine Kommunikation zum elektronischen Personalausweis benötigen, ist damit nicht ersichtlich, welche konkrete Technologie zum Einsatz kommt. NFC als Kommunikationsschnittstellen wurde in der Klasse `ISO14443NFC` implementiert (siehe Abbildung 5.11).

Paket: `de.tud.cdc.mecca.card`

- `de.tud.cdc.mecca.card.CardConfig`
  Die Klasse definiert die File Identifer der `MasterFile` und der Dateien `EF.CardAccess`, `EF.CardSecurity`, `EF.CVCA` usw. [3, Tabelle A.1, A.11].

- `de.tud.cdc.mecca.card.CardProxy`
  Die Klasse stellt die Methode *getEFCardAccess()* bereit, die eine Menge der ausgelesenen Security-Infos zurück gibt.

Paket: `de.tud.cdc.mecca.iso.ISO14443`

- `de.tud.cdc.mecca.iso.ISO14443.IISO14443`
  Das Interface definiert eine Methode *sendAPDU(AbstractCommandAPDU apdu)* zum Austausch von APDUs.

- `de.tud.cdc.mecca.iso.ISO14443.ISO14443NFC`
  Die Klasse implementiert das Interface `IISO14443` und stellt eine `ISO14443Connection` zur Kommunikation mittels NFC gemäß der JSR 257 Contactless Communication API [78] bereit. Die Methode *sendAPDU(AbstractCommandAPDU apdu)* ermöglicht das Senden bzw. Empfangen einer APDU über NFC. Die Antwort wird anhand ihres Statuscodes klassifiziert und als Instanz einer `ResponseNormalOperation` oder `ResponseError` zurückgegeben.

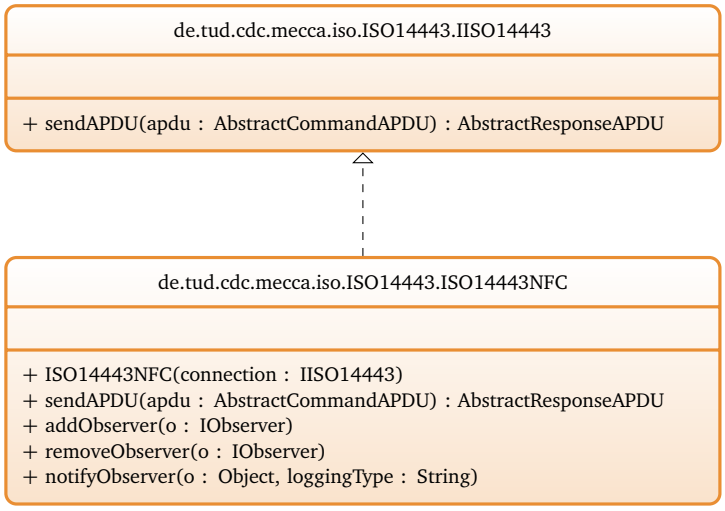

Abbildung 5.11.: Klassendiagramm: Kontaktlose Schnittstelle

5.3.5 Kryptographische Mechanismen

Die kryptographischen Komponenten, wie das Berechnen und Ableiten der Schlüssel, die das PACE Protokoll benötigt, wurden in die Klassen `CipherSuite` und `KDF` ausgelagert. Dadurch ist die Implementierung der Klasse `PACE` weitestgehend unabhängig von der konkreten Implementierung der kryptographischen Mechanismen und des verwendeten Cryptography Service Provider (CSP).

Paket: `de.tud.cdc.mecca.crypto`

- `de.tud.cdc.mecca.crypto.CipherSuite`
  Die Klasse `CipherSuite` (Abbildung 5.12) implementiert die kryptographischen Bestandteile des PACE Protokolls. Die Methode *decryptNonce(byte[] key, byte[] nonce)* entschlüsselt die vom elektronischen Personalausweis verschlüsselte Zufallszahl $s = D(K_\pi, E(K_\pi, s))$. Der Schlüssel wird von der Klasse `KDF` berechnet. Des Weiteren stehen mehrere Methoden *getKeyPair(...)* zur Verfügung, die die Schlüssel berechnen. Ohne Parameter liefert die Methode einen flüchtigen Schlüssel (Schritt 2 Abbildung 3.5). Bei der Verwendung mit den neuen Domain-Parametern wird der Schlüssel PK_{PICC} (Schritt 3) berechnet. Die dritte Variante der Methode wird jeweils in Schritt 2 und 3 verwendet. Der Parameter *data* repräsentiert dabei die vom Chip empfangenen Daten. Die Methode *getSecretKey(byte[] keyBytes)* berechnet die Sitzungsschlüssel K_{MAC} bzw. K_{ENC}. Das Berechnen des neuen Punktes auf der elliptischen Kurve [3, Kapitel A.3.4] implementiert die Methode *mapPoint(...)*.

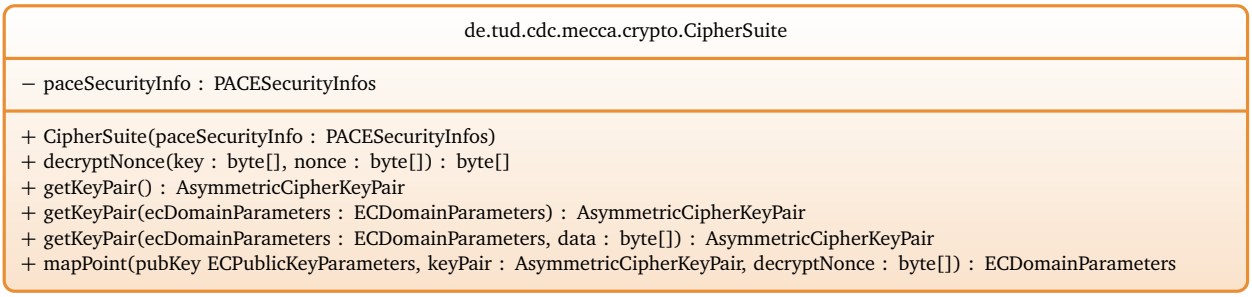

Abbildung 5.12.: Klassendiagramm: CipherSuite

- `de.tud.cdc.mecca.crypto.KDF`
  Die Klasse `KDF` (Abbildung 5.13) implementiert die Key Derivation Function [3, Kapitel A.2.3]. Bei der Instantiierung wird eine Referenz auf das verwendete `PACESecurityInfos` Objekt übergeben, aus dem die Klasse die Informationen über den verwendeten symmetrischen Verschlüsselungsalgorithmus (AES oder Triple-DES) bezieht. Es stehen unter anderem die Methoden *derivePI(byte[] secret)* zum Ableiten des Schlüssels K_π, *deriveMAC(byte[] secret)* zum Ableiten des Schlüssels K_{MAC} und *deriveENC(byte[] secret)* zum Ableiten des Schlüssels K_{ENC} zur Verfügung.

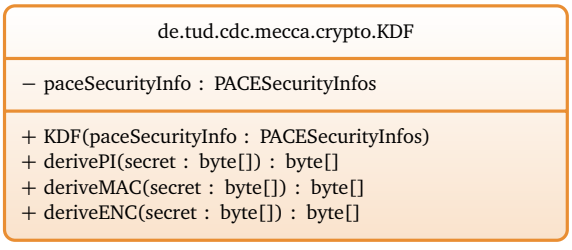

Abbildung 5.13.: Klassendiagramm: Key Derivation Function

5.3.6 PACE

Die Klasse implementiert die einzelnen Schritte des PACE Protokolls und deren Abfolge. Die kryptografischen Bestandteile, die Kommunikationsschnittstelle zum elektronischen Personalausweis und die APDUs wurden ausgelagert und bereits in den vorherigen Kapiteln erläutert. Dadurch entstand eine sehr übersichtliche und komprimierte Implementierung des Protokolls. Die fünf Schritte des Protokolls (siehe Kapitel 3.3.2) wurden in einzelnen Methoden implementiert und starten bei einer erfolgreichen Durchführung jeweils den nächsten Protokollschritt. Im Wesentlichen geschieht das, wenn von der kontaktlosen Schnittstelle eine Instanz der Klasse `ResponseNormalOperation` zurückgegeben wird. Nur die Methode `MSESetAT()` muss ggf. eine Instanz der Klasse `ResponseError` verarbeiten, falls beispielsweise das Passwort gesperrt oder blockiert ist oder die Initialisierung des PACE Protokolls fehlschlägt.

Paket: `de.tud.cdc.eac.pace`

- `de.tud.cdc.mecca.eac.protocols.PACE`
  Wie aus Abbildung 5.14 ersichtlich, implementiert die Klasse `PACE` das Interface `IObserable` und ist damit in der Lage, Statusinformationen an registrierte Objekte weiterzuleiten. Die Klasse `GUI` registriert sich beispielsweise, um Statusmeldungen auf der graphischen Oberfläche darzustellen. Instantiiert wird die Klasse `PACE` mit einem Objekt, welches das Interface `IISO14443` implementiert und demnach eine Methode `sendAPDU(...)` zur Kommunikation mit dem elektronischen Personalausweis bereitstellen muss und einer Instanz des Objekts `PACESecurityInfos`, das unter anderem die Informationen der Domain-Parameter enthält. Die Methode `run(password : byte[])` startet das PACE Protokoll. Dabei wird nur das Passwort übergeben, der Typ des Passwortes, d. h. PIN, CAN, usw. kann von dem Objekts `PACESecurityInfos` erfragt werden. Die berechneten Sitzungsschlüssel K_{MAC} und K_{ENC} werden über die Methoden `getKeyMAC()` bzw. `getKeyENC()` bereitgestellt.

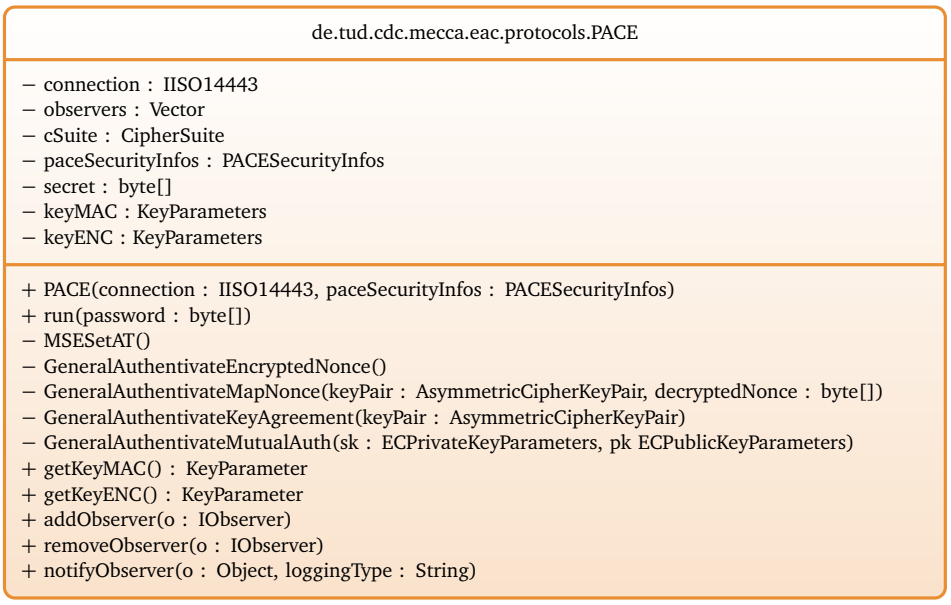

Abbildung 5.14.: Klassendiagramm: PACE

5.3.7 Weitere Funktionalitäten

Für die graphische Oberfläche und als Rückmeldung für den Benutzer ist es wichtig, dass Informationen über den aktuellen Status des PACE Protokolls zugänglich sind. Bei dem Design der Implementierung wurde darauf Wert gelegt, dass in dieser Hinsicht keine Abhängigkeiten zwischen den einzelnen Komponenten bestehen. Um dieses Ziel zu erreichen, wurde das Observer Pattern [79] verwendet. Die Funktion des Observer Pattern lässt sich beispielsweise wie folgt beschreiben:

> *„Allow objects to dynamically register dependencies between objects so that an object will notify those objects that are dependent on it when its state changes.“ — Mark Grand* [81]

Der Vorteil in der Verwendung dieses Entwurfmusters ist die geringe Kopplung der Komponenten. Die einheitliche Schnittstelle sorgt dafür, dass sofort ersichtlich ist, an welchen Stellen in der Implementierung Informationen ausgetauscht werden. Die gleiche Funktionalität ohne Observer Pattern, würde eine gegenseitige Referenzierung der jeweiligen Klassen bedeuten, was den Austausch bzw. Verzicht auf eine Komponente sehr schwierig macht. Um den Vorteil zu verdeutlichen, erläutern wir den Einsatz des Observer Pattern für die Klasse `PACE`. Um die graphische Benutzeroberfläche mit Statusmeldungen zu versorgen, hätte bei der Instantiierung der Klasse `PACE` eine Referenz der GUI übergeben werden müssen. Bei Statusmeldungen würden dann Methoden der GUI aufgerufen. Bei dieser Variante entsteht aber eine starke Kopplung zwischen den beiden Komponenten, d. h. die Klasse `PACE` benötigt eine Referenz auf eine graphische Oberfläche, auch wenn diese vielleicht nicht benötigt wird. Durch den Einsatz des Observer Pattern sind die Komponenten voneinander unabhängig. Damit die graphische Oberfläche Informationen erhalten kann, muss sie sich nur bei der Klasse `PACE` registrieren. Des Weiteren werden keine speziellen Methodenaufrufe benötigt und es können sich mehrere Klassen gleichzeitig registrieren. Infolgedessen ist es möglich, die PACE Implementierung auch auf einem Rechner einzusetzen, der beispielsweise keine oder eine andere graphische Oberfläche bereitstellt.
Die Implementierung ist angelehnt an die Version, die die Java Standard Edition zur Verfügung stellt, aber im Funktionsumfang der Java Micro Edition nicht enthalten ist. Die Struktur des Observer Pattern ist in Abbildung 5.15 abgebildet und in folgenden Klassen realisiert:

Paket: `de.tud.cdc.mecca.observer`

- `de.tud.cdc.mecca.observer.IObservable`
  Die Klasse stellt eine Schnittstelle für Objekte bereit, die Statusmeldungen erzeugen und von anderen Objekten beobachtet werden können. Mittels der Methoden *addObserver(IObserver o)* und *removeObserver(IObserver o)* kann sich ein Objekt registrieren (um Statusmeldungen zu erhalten) bzw. sich abmelden. Über die Methode *notiyObserver(Object o, String loggingType)* werden angemeldete Objekte informiert.

- `de.tud.cdc.mecca.observer.IObserver`
  Die Schnittstelle definiert die Methode *update(IObservable o, Object arg, String loggingType)* zum Benachrichtigen von Objekten. Durch die Referenz *IObservable o* ist ersichtlich von welchem Objekt die Informationen stammen. Somit besteht die Möglichkeit, Statusmeldungen von verschiedenen Quellen differenziert verwenden bzw. auswerten zu können. Das Objekt *arg* beinhaltet dabei die eigentliche Information bzw. Nachricht. Auf eine Spezifikation des Datentyps wurde bewusst verzichtet, um neben Informationen vom Typ String auch ganze Abbilder einer APDU in Form eines Byte-Arrays übermitteln zu können. Der Parameter *loggingType* definiert den Typ der Nachricht (siehe unten).

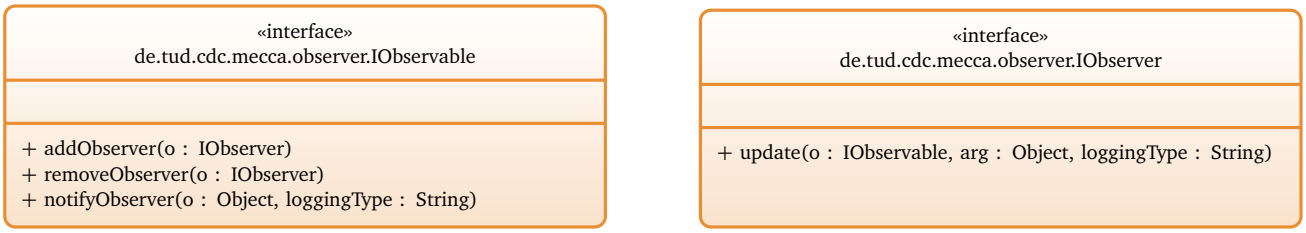

Abbildung 5.15.: Klassendiagramm: Observer

Bei der Entwicklung auf einem Handy steht keine Konsole zur Ausgabe von Informationen wie bei einer integrierten Entwicklungsumgebung (IDE) auf einem Rechner zur Verfügung. Informationen können auf einem Handy nur innerhalb einer graphischen Oberfläche angezeigt werden, was die Entwicklung erheblich erschwert. Weil das Anzeigen von vielen Informationen auf dem Display sehr unübersichtlich und begrenzt ist, wurde die Klasse `Logging` (Abbildung 5.16) entwickelt. Damit ist es möglich Informationen bzw. Meldungen aufzuzeichnen und abzuspeichern.

Paket: `de.tud.cdc.mecca.common`

- `de.tud.cdc.mecca.common.Logging`
  Die Klasse implementiert das Interface `IObserver` und speichert alle Statusmeldungen in der Datenstruktur eines Vektors. Zusätzlich wird zur jeder Statusmeldung ein Zeitstempel gespeichert. Die Methode *saveLogFile( )* speichert die gesammelten Statusmeldungen in einer Datei auf dem Handy. Der Dateiname sowie der Speicherort und die Dateiendung sind dabei variabel.

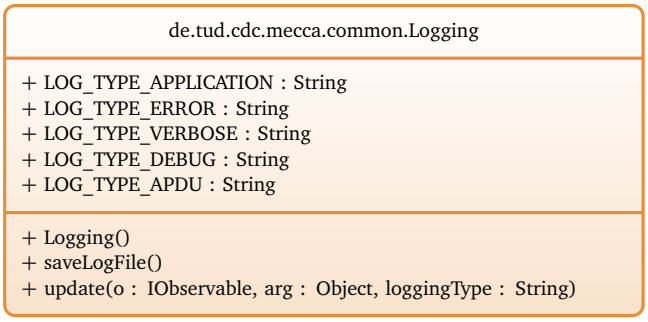

Abbildung 5.16.: Klassendiagramm: Logging

Durch den Einsatz des Observer Patterns bei der Klasse `Logging` ist es möglich nur bestimmte Statusmeldungen von Komponenten aufzeichnen zu lassen. Registriert sich die Klasse `Logging` beispielsweise nur bei der Klasse `PACE`, werden auch nur die Statusmeldungen von `PACE` aufgezeichnet. Weil Objekte ggf. zu viele bzw. zu detaillierte Statusmeldungen senden, wurden zusätzlich verschiedene Typen definiert:

- **LOG_TYPE_APPLICATION**
  Meldungen von diesem Typ dienen ausschließlich zur Interaktion zwischen den Komponenten. PACE informiert beispielsweise die graphische Oberfläche mit diesem Typ von Statusmeldungen über den aktuellen Protokollschritt.

- **LOG_TYPE_ERROR**
  Wird beim Auftreten von Fehlern im Protokollablauf usw. verwendet.

- **LOG_TYPE_VERBOSE**
  Statusmeldungen vom Typ Verbose geben einen detaillierteren Aufschluss über die Aktionen des Programms. Als Beispiel sei hier das Auswählen und Auslesen des `EF.CardAccess` genannt.

- **LOG_TYPE_DEBUG**
  Der Typ Debug liefert noch detailliertere Informationen, wie beispielsweise die berechneten Schlüssel.

- **LOG_TYPE_APDU**
  Statusmeldung von Typ APDU beinhalten ein komplettes Abbild der versendeten und empfangenen APDUs.

Unter Verwendung der Observer Schnittstelle zum Informationsaustausch zwischen den einzelnen Komponenten lassen sich auch bequem Aufzeichnungen (Logs) und Timings erstellen. Die Klasse `Logging` fungiert dabei als Observer und speichert die Statusmeldungen aller registrierten Klassen mit Datum und Uhrzeit in einer Logfile, die zusätzlich auf dem Handy abgelegt und zu einem späteren Zeitpunkt ausgewertet werden kann. Durch den Zeitstempel der jeweiligen Einträge können Timings erstellt werden, d. h. Informationen über die Laufzeit bzw. Aufwand der einzelnen Protokollschritte gesammelt und ausgewertet werden. Dabei könnten auch Vergleiche zwischen den jeweiligen kryptographischen Verfahren wie Data Encryption Standard (DES) oder Advanced Encryption Standard (AES) in Verbindung mit DH (Diffie-Hellman) bzw. ECDH (Elliptic Curve Diffie-Hellman) erstellt werden, wobei sich diese ausschließlich auf die Performance beziehen würden und nicht auf die Sicherheit dieser kryptographischen Verfahren.
Mit den Einsatz der Klasse `Logging` und der verschiedenen Typen von Statusmeldungen steht ein flexibles und universal einsetzbares Logging-Framework zur Verfügung. Der Einsatz des Observer Patterns erlaubt es, die Komponenten unabhängig voneinander zu entwickeln und auszutauschen.

5.4 Analyse

In Anbetracht der geringen Systemressourcen eines mobilen Gerätes ist vor allem die Laufzeit des Programms interessant. Um jedoch quantitative Aussagen über die Messwerte treffen zu können, wurden Referenzwerte mit einem Rechner und Kartenleser erstellt. Die Messwerte wurden an folgenden Abschnitten der Implementierung erfasst:

1. Extrahieren der SecurityInfos
   - 1.1 Selektieren der `MasterFile` und der `EF.CardAccess` mittels SELECT FILE APDU
   - 1.2 Auslesen der EF.CardAccess mittels READ BINARY APDU
   - 1.3 Generieren der PACESecurityInfos
2. PACE
   - 2.1 MSESetAT
   - 2.2 General Authentivate Encrypted Nonce
   - 2.3 General Authentivate Map Nonce
   - 2.4 General Authentivate Key Agreement
   - 2.5 General Authentivate Mutual Authentication

Die Werte wurde jeweils nach den einzelnen Punkten erhoben und in Millisekunden gemessen. Die Messpunkte bei PACE sind die Methoden (Abbildung 5.14), die die fünf Schritte (Abbildung 3.5) des Protokolls implementieren. Die Messwerte für das Nokia 6212 sind in Tabelle 5.17 abgebildet. Tabelle 5.18 beinhaltet die Messwerte, die mit einem Rechner und einem Kartenleser erhoben wurden. Um einen zuverlässigen Mittelwert zu erhalten, wurden zehn Programmdurchläufe analysiert. Die einzelnen Spalten der Tabellen repräsentieren die Durchläufe und die Zeilen die jeweiligen Messpunkte mit den ermittelten Werten.

	1	2	3	4	5	6	7	8	9	10	∅
1.1	92	90	90	89	89	89	90	90	89	89	89,7
1.2	280	279	278	271	272	276	6228	278	276	276	871,4
1.3	612	612	611	606	608	610	6564	614	612	612	1206,1
2.1	1187	1200	1195	1178	1186	1182	11964	1189	1184	1184	2264,9
2.2	1323	1344	1326	1306	1325	1312	12110	6181	1320	1318	2886,5
2.3	23645	23717	23345	23190	23636	23326	34548	28298	24056	23610	25140,1
2.4	35902	36020	35580	35212	35580	36133	46753	40299	36155	35567	37320,1
2.5	41827	42028	41552	41159	41328	42118	52735	47081	42052	51364	43324,4
Gesamt	41827	42028	41552	41159	41328	42118	52735	47081	42052	51364	

Abbildung 5.17.: Messwerte: Nokia 6212

Wie aus Tabelle 5.17 ersichtlich, steigen die Werte der Messpunkte zwischen 1.1 bis 2.2 recht gleichmäßig an. Das Verarbeiten der ASN.1 Datenstruktur und das Extrahieren der PACESecurityInfos geschieht im Durchschnitt in ca. 3 - 4 Sekunden. Auffällig sind außerdem die Werte in Punkt 2.3. An dieser Stelle werden die neuen Domain-Parameter berechnet, was im Durchschnitt in ca. 22 Sekunden erfolgt. Der Schlüsselaustausch (Punkt 2.4) benötigt im Durchschnitt ca. 12 Sekunden und die gegenseitige Authentifizierung (Punkt 2.5) ca. 6 Sekunden.

Bei den zehn Durchläufen sticht insbesondere der Siebte heraus, dort dauerte das Auslesen der Datei `EF.CardAccess` fast 6 Sekunden. Bei den Funktionstests des Prototypen war bereits auffällig, dass es immer wieder zu Verzögerungen kam, bis das Handy den elektronischen Personalausweis erkannt hatte und eine Verbindung aufbauen konnte, was ggf. durch die geringe Sendeleistung der NFC Schnittstelle des Nokia 6212 zu erklären ist. Unterschiedliche Positionen zwischen Handy und ePA sorgten teilweise auch dafür, dass keine Verbindung oder nur eine sehr langsame zustande kam, wie es im siebten Durchlauf der Fall war.

	1	2	3	4	5	6	7	8	9	10	∅
1.1	251	230	241	190	200	230	180	220	231	220	219,3
1.2	321	310	321	260	280	310	250	300	311	300	296,3
1.3	401	370	381	330	340	380	300	350	371	370	359,3
2.1	631	570	611	520	530	601	511	551	571	600	569,6
2.2	711	641	691	600	631	681	591	621	651	671	648,9
2.3	1222	1101	1222	1101	1101	1222	1061	1122	1162	1151	1146,5
2.4	1502	1402	1532	1412	1392	1522	1342	1402	1462	1442	1441
2.5	1603	1502	1633	1512	1482	1622	1432	1502	1552	1542	1548,2
Gesamt	1603	1502	1633	1512	1482	1622	1432	1502	1552	1542	

Abbildung 5.18.: Messwerte: Rechner und Kartenleser

Betrachten wir jetzt die Referenzwerte aus Tabelle 5.18: Bei einer durchschnittlichen Gesamtlaufzeit von ca. 1,5 Sekunden ist diese deutlich geringer als beim Nokia 6212. Auffällig an den Messwerten des Rechners im Vergleich zu denen des Handys sind nicht nur die höheren Werte, die durch die geringen Systemressourcen entstehen, sondern auch die deutlich schnellere Kommunikation. Betrachten wir den ersten Durchlauf: In diesem benötigt das Handy ca. 190 Millisekunden zum Auslesen der Datei `EF.CardAccess`, der Rechner bzw. das stationäre Lesegerät hingegen nur 70 Millisekunden.
Analysieren wir die Werte des rechenintensiven Punktes 2.3, dann benötigt der Rechner im Durchschnitt ca. 5 Sekunden, was ca. 32 % der Gesamtlaufzeit bedeutet. Bei dem Handy tragen die ca. 22 Sekunden für die Berechnung der neuen Domain-Parameter jedoch 51 % zur Gesamtlaufzeit bei. Insbesondere bei den Berechnungen innerhalb des PACE Protokolls machen sich die geringen Systemressourcen des Handys bemerkbar. Ein weiteres Beispiel ist das Generieren der PACESecurityInfos. Bei den Messungen im ersten Durchlauf benötigte das Handy 332 Millisekunden, der Rechner hingegen nur 80 Millisekunden.

Anmerkung: Die Zeitmessungen wurden mir Hilfe der Klasse `Logging` erstellt, d. h. nicht direkt innerhalb der Methoden, was zu Abweichungen bzw. einer höheren Laufzeit in Folge von zusätzlichen Methodenaufrufen führt. Die Messwerte aus Tabelle 5.18 wurden auf einem Rechner mit einem Intel Core 2 Quad Q8300 (4x 2.50GHz) und 4 GB RAM sowie dem Chipkartenleser SDI010 von SCM Microsystems erstellt. Informationen über die Hardwarekomponenten des Nokia 6212 sind nicht verfügbar.

5.5 Zusammenfassung

Der Prototyp wurde auf Basis der Java Mirco Editon entwickelt. Als NFC-fähiges Handy wurde das Nokia 6212 classic verwendet. Für eine übersichtliche Struktur wurden die implementierten Klassen nach ihrer Funktionalität in Pakete eingeordnet und Design Pattern für einen hohen Qualitätsstandard benutzt. Die wesentlichen Komponenten des Prototyps sind die Implementierungen der APDUs, der ASN.1 Datenstrukturen, der graphischen Benutzeroberfläche sowie das PACE Protokoll und der dazugehörigen kryptographischen Komponenten.
Neben der Erweiterbarkeit stand bei der Entwicklung ebenso die Wartbarkeit im Vordergrund. Bis zur flächendeckenden Einführung des elektronischen Personalausweises sind weitere Änderungen an der Extended Access Control Spezifikation nicht ausgeschlossen. Das Ziel, eine flexible und leicht zu modifizierende Software zu entwickeln, wurde auch deshalb motiviert, weil nicht alle kryptographischen Mechanismen implementiert werden konnten. Die Implementierung des Prototypen ist dahingehend ausgelegt, dass die weiteren, laut EAC Spezifikation einsetzbaren bzw. möglichen, kryptographischen Mechanismen komfortabel hinzugefügt werden können.
Bei der Entwicklung wurde insbesondere großen Wert auf eine starke Abstraktion und geringe Kopplung der Komponenten gelegt. Dadurch können Veränderungen leicht umgesetzt werden und einzelne Komponenten ausgetauscht bzw. ersetzt werden, weil diese voneinander unabhängig sind und eine einheitliche und klar definierte Schnittstelle zur Kommunikation nutzen. Die Abstraktion sorgt für eine strukturierte Implementierung des PACE Protokolls, da beispielsweise der konkrete Aufbau der APDUs und Statuswörter ausgelagert und zentral definiert wurde.
Für die Laufzeitmessung wurde das entwickelte Logging-Framework verwendet. Dadurch war es möglich, ohne Veränderungen an den implementierten Komponenten, die Laufzeit des Prototyps komfortabel zu bestimmen. Die Messungen ergaben eine höhere Laufzeit, als auf einem Rechner mit stationärem Lesegerät, was mit den beschränkten Systemressourcen und der geringeren Sendeleistung der NFC Schnittstelle des Handys zu begründen ist.
Die auf einem mobilen Gerät verfügbare Java Micro Edition zeigt insbesondere bei den zur Verfügung stehenden kryptographischen Mechanismen ihre Schwächen. In diesem Bereich steht nur ein Bruchteil der in der Java Standard Edition bereitgestellten Funktionalität zur Verfügung, was den Einsatz eines externen Cryptography Service Provider unausweichlich macht. Demzufolge entsteht eine an den Provider angepasste Implementierung, was durch das Auslagern der benötigten kryptographischen Funktionalitäten in eigene Klassen bestmöglich kompensiert wurde.
Die eingeschränkte Funktionalität der Java Micro Edition spiegelt sich auch in mehreren anderen Designentscheidungen wieder. Eine Implementierung des Observer Pattern und die Basisklassen der APDUs sind bei der Java Standard Version bzw. der JavaCard API auf einem Rechner verfügbar, auf einem mobilen Endgerät jedoch nicht. Daher mussten diese zusätzlich implementiert werden, was über die eigentliche Implementierung des PACE Protokolls hinausgeht. Diese zusätzliche Funktionalität verschafft dem Prototypen aber ein robusteres Design, was eine verständlichere und übersichtlichere Struktur sowie eine bessere Erweiterbarkeit und Wiederverwendbarkeit zur Folge hat.

6 Perspektiven

Im folgenden Abschnitt möchten wir einen Ausblick geben, der Thesen, Möglichkeiten und Vorstellungen zukünftiger Projekte erläutert, die im Bereich des elektronischen Personalausweises und der NFC-Technologie angesiedelt sind.

Anwendungsszenarien

Der elektronische Personalausweis eröffnet mit seinen Funktionen des elektronischen Identitätsnachweises und der qualifizierten elektronischen Signatur eine Vielzahl neuer Anwendungsmöglichkeiten, die wir ausführlich in Kapitel 2.3 erläutert haben. Die Schutzziele der Vertraulichkeit, Integrität, Verfügbarkeit und Authentizität stehen dabei im Vordergrund und werden durch Extended Access Control realisiert. Der ePA wird eine zuverlässige und glaubwürdige Authentifizierung des Inhabers bieten und demnach auch für Bereiche Verwendung finden, in denen ein Identitätsnachweis von größter Bedeutung ist, wie beispielsweise bei einer politischen Wahl.
Die Idee elektronische Systeme bei Wahlen einzusetzen, existiert seit geraumer Zeit und wird in einigen Ländern bereits praktiziert. Kritische Stimmen und deren Bestätigung durch theoretisch manipulierbare Wahlcomputer sowie klare Verstöße gegen das Wahlrecht ließen die Technik teilweise wieder verschwinden. Die elektronische Datenverarbeitung beschränkt sich dabei jedoch ausschließlich auf das Erfassen der einzelnen Stimmen der Wähler und nicht auf den gesamten Prozess. Dass sich der Ablauf einer Wahl, das heißt das Identifizieren bzw. die Authentifizierung des Wählers und die Erfassung seiner Stimme, vollständig elektronisch realisieren lässt, verdeutlicht der Artikel *eVoting with the European Citizen Card* [82]. Auf Grundlage der Technologie und der Infrastruktur des elektronischen Personalausweises und der in Extended Access Control zusammengefassten Sicherheitsprotokolle kann mit dem im Artikel vorgestellten Verfahren eine sichere und authentische Wahl vom privaten Computer aus realisiert werden. Die Sicherheitsbedenken, die bei dem Einsatz von Wahlcomputern immer wieder aufkommen, sind mögliche Manipulationen der Geräte bzw. Ergebnisse durch Personen, die Zugang zu den Wahlcomputern haben. Ein eVoting bzw. eine elektronische Wahl über das Internet würde eine zentrale Stelle schaffen, die effizienter zu kontrollieren ist, aber auch einen klassischen „Single Point of Failure“ darstellen. Langfristig gesehen werden durch die Ausweispflicht alle Bürgerinnen und Bürger der Bundesrepublik Deutschland über 16 Jahre einen elektronischen Personalausweis besitzen. Laut Pressemitteilung[1] vom 15.05.2007 des statistischen Bundesamtes gab es in 80,6 % der Privathaushalte im Jahr 2006 mindestens ein Mobiltelefon. Die Verwendung eines NFC-fähigen Handys als Schnittstelle bzw. Lesegerät zwischen Computer und elektronischem Personalausweis, stellt, in Anbetracht der starken Verbreitung mobiler Geräte, ein realistisches Szenario dar. Darüber hinaus könnte der entwickelte Prototyp um das im Artikel [82] vorgestellte Verfahren erweitert werden und eine Wahl per Mobiltelefon ermöglichen. Die Stärke dieses Verfahrens liegt in den bereits weit verbreiteten und etablierten Technologien und einer sicheren und authentischen Identifizierung der Wähler. Eine Altersprüfung, das heißt, ob eine Person bereits das 17. Lebensjahr vollendet hat (und zusätzlich im Wählerverzeichnis eingetragen ist) und damit wahlberechtigt ist, kann ebenfalls durch die Funktion des elektronischen Personalausweises realisiert werden. Eine sichere und vertrauliche Wahl per mobilem Endgerät würde eine Stimmabgabe ortsunabhängig machen und so ggf. die Wahlbeteiligung erhöhen, darüber hinaus erhebliche Personal- und Verwaltungskosten einsparen.

[1] http://www.destatis.de/jetspeed/portal/cms/Sites/destatis/Internet/DE/Presse/pm/zdw/2007/PD07__019__p002.psml

Die Option, dass ein mobiles Gerät mit einem elektronischen Personalausweis kommunizieren kann, eröffnet weitere Anwendungsszenarien in sensiblen Bereichen. Bei einem Notruf könnten die personenbezogenen Daten des ePA direkt übertragen werden und so eine schnelle und zuverlässige Datenerfassung durch die Leitstelle ermöglichen. Der Notruf sieht sich seit geraumer Zeit dem Missbrauch ausgesetzt. Mit der *Verordnung über Notrufverbindungen*[2] ist der Notruf nur noch mit SIM-Karte gestattet, bisher war dies auch ohne SIM-Karte möglich. Durch eine Kopplung an den elektronischen Personalausweis könnte diese Option auch weiterhin aufrecht erhalten werden.

Die von der Daimler AG entwickelte *Keyless GO* Technik ermöglicht das Aufschließen und Starten des Fahrzeugs, ohne den Autoschlüssel aktiv zu benutzen. Eine Integration der NFC-Technologie im Fahrzeug könnte eine Kommunikation zum elektronischen Personalausweis herstellen und so den Fahrzeughalter identifizieren und die gleiche Funktionalität wie die *Keyless GO* Technik bieten. Zusätzlich könnten verschiedene Einstellungen, wie die Sitzposition, der bevorzugte Radiosender und die Temperatur der Klimaanlage, automatisch an den jeweiligen Fahrer angepasst werden.

Die Kombination aus der NFC-Technologie und dem elektronischen Personalausweis könnte ebenfalls neue Maßstäbe im Bereich der mobilen Bezahl-Funktionen setzen, dem sogenannten M-Payment (Mobile Payment) [83][84]. In diesem Bereich bietet die Deutsche Bahn AG ihren Kunden das Handy-Ticket[3] und die Stadt Darmstadt ermöglicht den Autofahrern auf ihren fast 800 öffentlichen Parkplätzen den Kauf[4] eines Parkscheins mit Hilfe eines Mobilfunktelefons. Allgemein und im Falle dieser beiden Beispiele ist jedoch eine Registrierung vorab notwendig, d. h., ein Kunde kann den Service nicht kurzfristig nutzen, sondern muss sich erst bei dem Service-Anbieter anmelden. Dabei entsteht bei jedem Anbieter eine weitere partielle Identität, die verwaltet und geschützt werden muss. In Verbindung mit dem elektronischen Personalausweis könnte den Service-Anbietern ein sicherer und glaubwürdiger Identitätsnachweis geboten werden, bei dem darüber hinaus auch direkt die Anschrift des Ausweisinhabers übermittelt werden kann, was für Abrechnungszwecke unausweichlich ist. Dieses System würde die Nutzung des Angebots unabhängig von einer Anmeldung machen bzw. eine spontane Nutzung ermöglichen und ggf. neue Kunden akquirieren.

[2] http://www.gesetze-im-internet.de/notrufv/index.html
[3] http://mobile.bahn.de
[4] http://www.parken-per-handy.de

Extended Access Control

Das in Kapitel 3.3.2 vorgestellte und in diesem Dokument fokussierte Password Authenticated Connection Establishment (PACE) Protokoll bildet den Grundstein für das Sicherheitskonzept Extended Access Control (EAC) des elektronischen Personalausweises. Der entwickelte Prototyp implementiert das PACE Protokoll und ermöglicht eine Erweiterung um die restlichen Sicherheitsprotokolle des Extended Access Control. Auf Grundlage des Prototyps ist die Implementierung der Terminal-Authentifizierung und Chip-Authentifizierung der nächste Schritt für eine vollständige EAC Implementierung auf einem NFC-fähigen mobilen Endgerät. Das PACE Protokoll garantiert eine sichere und integere Verbindung zwischen elektronischem Personalausweis und einem Mobiltelefon. Um sicherzustellen, dass mit einem authentischen ePA kommuniziert wird, erfordert es jedoch die Chip-Authentifizierung.
Insbesondere die Restricted Identification [3, Kapitel 4.5] Funktion des elektronischen Personalausweises könnte für eine mobile Anwendung interessant sein, falls sie ohne eine komplette vorherige Ausführung von EAC möglich wäre. Die Erweiterung des Prototyps um diese Funktion würde es dem mobilen Gerät ermöglichen, einen ePA wiederzuerkennen, ohne erneut auf die personenbezogenen Daten des elektronischen Personalausweises zuzugreifen. Wie in Kapitel 4.1.5 erläutert, unterstützen die in Kaptiel 4.1 vorgestellten Betriebssysteme für mobile Geräte keine Mehrbenutzersysteme. Unterschiedliche bzw. an die Bedürfnisse angepasste Einstellungen und verschiedene Benutzerkonten sind folglich nicht möglich. Mit einer Authentifizierung gegenüber dem Mobiltelefon mittels elektronischem Personalausweis und der Restricted Identification Funktion könnten Mehrbenutzersysteme für mobile Geräte effizient und sicher realisiert werden. Restricted Identification würde eine schnelle und benutzerfreundliche Authentifizierung ermöglichen und das klassische Problem durch das Vergessen der PIN lösen. Ein Mobiltelefon könnte anhand des jeweiligen Personalausweises benutzerdefinierte Einstellungen laden, aber auch diverse Funktionen regulieren, wie die ausschließliche Wahl definierter Rufnummern. Insbesondere mobile Geräte, die im geschäftlichen Umfeld durch den Arbeitgeber bereitgestellt werden und demnach mehrmals den Benutzer wechseln, stellt diese Option der Personalisierung und Zugriffskontrolle eine effiziente Möglichkeit dar. Für die Nutzer ergibt sich der Schutz der gewählten Einstellungen, aber auch der ggf. persönlichen auf dem Gerät gespeicherten Daten, und andere Nutzer können Einstellungen nicht verändern oder Daten einsehen können.

Graphische Benutzeroberfläche

Der entwickelte Prototyp basiert auf den Graphical User Interface (GUI) Komponenten der Java Micro Edition. Dadurch entstehen Einschränkungen und ein geringer Spielraum für die Entwicklung eines modernen Designs der Anwendung. Das tatsächliche Erscheinungsbild der Software ist ggf. bei jedem Gerät unterschiedlich und vom Hersteller abhängig. Um eine benutzerfreundlichere und optisch ansprechende graphische Oberfläche zu entwickeln, müssten externe und teilweise kommerzielle Bibliotheken verwendet werden. Eine Auswahl vorhandener Bibliotheken zur Entwicklung professioneller graphischer Oberflächen sind im Folgenden aufgeführt:

- J2ME Polish[5]
  Eine große Sammlung von Werkzeugen zur Entwicklung von J2ME-Anwendungen stellt J2ME Polish bereit. Neben einem Build-Tool, Logging Framework und zahlreichen UI-Elementen werden auch Cascading Style Sheets (CSS) unterstützt. J2ME Polish ist unter der GPL[6] bzw. einer kommerziellen Lizenz verfügbar. Detaillierte Informationen stellt der Artikel [85] zur Verfügung.

- J4ME[7]
  Java For Me (J4ME) zeichnet sich insbesondere durch die Global Positioning System (GPS) Unterstützung und das Logging-Framework aus. Weitere UI-Element, wie Fortschrittsleiste, Themes, Splash Screens usw., werden ebenfalls unterstützt. J4ME wird unter der Apache-Lizenz 2.0 vertrieben.

- LWUIT[8]
  Das von SUN entwickelte Lightweight UI Toolkit (LWUIT) ist unter der GPL Version 2 lizenziert. Angelehnt an das Framework SWING sollen die graphischen Oberflächen auf allen Geräten ähnlich sein. Unterstützt werden unter anderem Layouts, Themes, Fonts, SVG und Animationen.

Unter Verwendung dieser Bibliotheken könnte für den Prototyp ein einheitliches und professionelles Erscheinungsbild entworfen und umgesetzt werden. Unabhängig von dem jeweiligen Gerät wäre ein identisches Design realisierbar, was insbesondere für einen *produktiven Einsatz* von Bedeutung ist. Ein Designvorschlag ist in Abbildung 6.1 illustriert.

Abbildung 6.1.: GUI Designvorschlag

[5] http://www.j2mepolish.org
[6] http://www.gnu.org/licenses/licenses.html
[7] http://code.google.com/p/j4me/
[8] https://lwuit.dev.java.net

eCard-API-Framework

Das eCard-API-Framework [86] stellt eine einheitliche Schnittstelle für die Chipkarten-Projekte der Bundesregierung zur Verfügung. Das Ziel ist es, eine einfache und plattformunabhängige Kommunikation zwischen den einzelnen Anwendungen und den jeweiligen Chipkarten bereitzustellen, um so einen einheitlichen Zugang auf die einzelnen Funktionen der Chipkarten-Projekte zu ermöglichen. Der Aufbau des eCard-API-Frameworks ist in mehrere Schichten unterteilt, die den Zugriff auf die Funktionen, die Verwaltung und Nutzung elektronischer Identitäten, kryptographische und biometrische Mechanismen sowie die Kommunikation mit der Chipkarte über die jeweilige Schnittstelle realisieren. Ausführliche Informationen stellen die Artikel [87] und [88] bereit. Der Fokus der eCard-Strategie liegt auf der elektronischen Authentifizierung und der qualifizierten elektronischen Signatur und umfasst insbesondere folgende Projekte:

- Elektronische Gesundheitskarte (eGK)
- Elektronischer Personalausweis (ePA)
- Elektronischer Reisepass (ePass)
- Elektronische Steuererklärung (ELSTER)
- Elektronischer Einkommensnachweis (ELENA)

Der entwickelte Prototyp könnte erweitert bzw. modifiziert werden, um die definierten Schnittstellen des eCard-API-Framework zur Verfügung zustellen. Demzufolge könnten Bestandteile des Prototyps, wie die Kommunikation zu einer Chipkarte, für weitere Projekte der eCard-Strategie genutzt bzw. wiederverwendet werden. Der Prototyp würde so eine einheitliche Schnittstelle auf Basis eines NFC-fähigen mobilen Gerätes zugänglich machen, die sich im Terminal Layer (IFD-Interface) [86, Teil 6] des eCard-API-Frameworks ansiedeln würde. Dieses Konzept würde eine Kommunikation zwischen einem NFC-fähigen Gerät und beispielsweise der elektronischen Gesundheitskarte (eGK) bzw. den weiteren Chipkarten-Projekten realisieren. Zu beachten ist, dass die eGK als kontaktbehaftete Chipkarte geplant ist. Um eine Kommunikation mit einem mobilen Gerät aufzubauen, müssten die Geräte über einen Magnetstreifen-Lesegerät verfügen, oder spezielle Kartenhüllen entwickelt werden, die die Informationen der Karte per ISO/IEC 14443 [14] Schnittstelle zugänglich machen. Bürgerinnen und Bürger (mit einem NFC-fähigen Gerät) könnten dann nicht nur die in Kapitel 2.3 vorstellten Anwendungsszenarien nutzen, sondern auch im Internet zugängliche Angebote in Verbindung mit ihrer elektronischen Gesundheitskarte. Realisierbar wäre der Einblick in die eigene Krankenakte oder Terminvereinbarungen bei einem Facharzt, bei dem ggf. Informationen aus der Krankenakte erforderlich sind. Bei der elektronischen Gesundheitskarte ist unter anderem geplant, Rezepte digital zu speichern. Von einem Arzt werden diese elektronischen Rezepte auf der eGK gespeichert und in einer Apotheke ausgelesen. Im Geschäftsbereich einer Online-Apotheke müssen derzeit Rezepte auf dem Postweg an die Apotheke geschickt werden. Ein NFC-fähiges Mobiltelefon könnte als Schnittstelle zwischen der elektronischen Gesundheitskarte und dem Onlineportal bzw. Onlineshop einer Apotheke fungieren und den Versicherten die Möglichkeit bieten, ihre Rezepte in elektronischer Form direkt an die Apotheke zu übermitteln. Diese Option würde nicht nur eine Kostenersparnis für die Versicherten bieten, sondern auch die Möglichkeiten der Fälschung eines Rezeptes minimieren.

XML

Für die Implementierung des eCard-API-Framework wird ein Extensible Markup Language (XML) Parser benötigt. Die JavaME Version bietet diese Funktionalität nicht standardmäßig. Wie im Artikel *Parsing XML in J2ME* [89] beschrieben stehen dazu mehrere externe XML Parser zur Verfügung. Die Verwendung des XML Formats würde sich auch für das erstellen der Logfile eignen und somit ein standardisiertes und plattform-unabhängiges Dateiformat verwenden.

Zusammenfassung

Zusammenfassend lässt sich feststellen, dass der entwickelte Prototyp die Basis für weitere Projekte im Bereich der eCard-Strategie bzw. einer Mobile European Citizen Card Application (MECCA) bildet. Der nächste Schritt wäre die Implementierung der Terminal-Authentifizierung, Chip-Authentifizierung und der Restricted Identification, also eine vollständige Umsetzung der Extended Access Control sowie eine professionelle graphische Oberfläche und die Bereitstellung der Schnittstellen gemäß des eCard-API-Frameworks.

A Anhang

A.1 MRZ-Daten

Die MRZ-Daten (maschinenlesbare Zone) bestehen nach §1 Abs. 3 PersAuswG aus folgenden Informationen:

1. Die Abkürzung *IDD* für Identitätskarte der Bundesrepublik Deutschland
2. Familienname(n)
3. Vorname(n)
4. Seriennummer (Behördenkennzahl, Ausweisnummer)
5. Abkürzung *D* für die Eigenschaft als Deutscher
6. Geburtsdatum
7. Gültigkeitsdauer
8. Prüfziffern
9. Leerstellen

QUELLE: Einführung des elektronischen Personalausweises in Deutschland [4]

A.2 Sicherheitsmerkmale des (alten) Personalausweises

Abbildung A.1.: Sicherheitsmerkmale des Personalausweises (bis Nov. 2010)

1. Identigram®: Holographisches Portrait
2. Identigram®: 3D-Bundesadler
3. Identigram®: Kinematische Bewegungsstrukturen
4. Identigram®: Makroschrift und Mikroschrift
5. Identigram®: Kontrastumkehr

6. Identigram®: Holographische Wiedergabe der maschinenlesbaren Zeilen
7. Identigram®: Maschinell prüfbare Struktur
8. Oberflächenprägung
9. Sicherheitsdruck mit mehrfarbigen Guillochen
10. Laserbeschriftung
11. Wasserzeichen

QUELLE: Bundesdruckerei GmbH

A.3 Chip-Authentifizierung

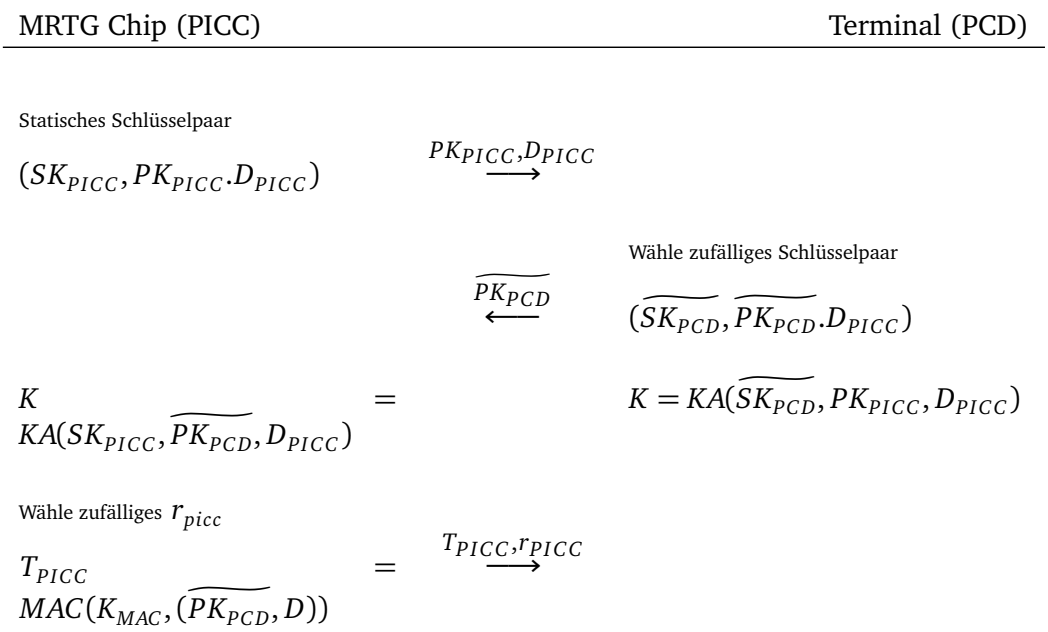

Abbildung A.2.: Chip-Authentifizierung (EAC Version 2.01) [3, Kapitel 4.3]

A.4 Terminal-Authentifizierung

MRTG Chip (PICC)		Terminal (PCD)
Statisches Schlüsselpaar $(SK_{PICC}, PK_{PICC}, D_{PICC})$	$\xrightarrow{PK_{PICC}, D_{PICC}}$	
	$\xleftarrow{\widetilde{PK_{PCD}}}$	Wähle zufälliges Schlüsselpaar $(\widetilde{SK_{PCD}}, \widetilde{PK_{PCD}}.D_{PICC})$
$K = KA(SK_{PICC}, \widetilde{PK_{PCD}}, D_{PICC})$		$K = KA(\widetilde{SK_{PCD}}, PK_{PICC}, D_{PICC})$
Wähle zufälliges r_{picc} $T_{PICC} = MAC(K_{MAC}, (\widetilde{PK_{PCD}}, D))$	$\xrightarrow{T_{PICC}, r_{PICC}}$	

Abbildung A.3.: Terminal-Authentifizierung (EAC Version 2.01) [3, Kapitel 4.4]

A.5 Restricted Identification

MRTG Chip (PICC)		Terminal (PCD)
Statisches Schlüsselpaar (SK_{ID}, PK_{ID})		Sector Public Key PK_{Sector}
	$\xleftarrow{PK_{Sector}, D}$	
$I_{ID}^{Sector} = H(KA(SK_{ID}, PK_{Sector}, D))$	$\xrightarrow{I_{ID}^{Sector}}$	

Abbildung A.4.: Restricted Identification (EAC Version 2.01) [3, Kapitel 4.5]

Abkürzungsverzeichnis

AES	Advanced Encryption Standard
APDU	Application Protocol Data Unit
AT	Authentication Template (MSE:Set AT Kommando)
BAC	Basic Access Control
CAN	Card Access Number
CSS	Cascading Style Sheets
DES	Data Encryption Standard
DH	Diffie-Hellman
EAC	Extended Access Control
ECDH	Elliptic Curve Diffie-Hellman
eGK	Elektronische Gesundheitskarte
EHUG	Elektronisches Handels- und Genossenschaftsregister
ELENA	Elektronischer Entgeltnachweis
ELSTER	Elektronische Steuererklärung
ePA	Elektronischer Personalausweis
ePass	Elektronischer Reisepass
ePerso	Elektronischer Personalausweis
GPL	GNU General Public License
GPS	Global Positioning System
GSM	Global System for Mobile Communications
GUI	Graphical User Interface
HSDPA	High Speed Downlink Packet Access
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
ICAO	International Civil Aviation Organization
ICC	Integrated Circuit Card, Smartcard, Chipkarte
IDE	Integrated Development Environment

ISO	International Organization for Standardization
ITU	International Telecommunication Union
J4ME	Java For Me
JavaME	Java Micro Edition
JSR	Java Specification Request
Keypad	Ziffernblock, Tastenfeld, Nummernblock
LWUIT	Lightweight UI Toolkit
MECCA	Mobile European Citizen Card Application
MRZ	Machine Readable Zone
MSE	Manage Security Environment (MSE:Set AT Kommando)
NFC	Near Field Communication
NIST	National Institute of Standards and Technology
PACE	Password Authenticated Connection Establishment
PAN	Personal Area Network
PASC	Password Authenticated Secure Channel
PCD	Proximity Coupling Device
PDA	Personal Digital Assistant
PersAuswG	Gesetz über Personalausweise
PICC	Proximity Integrated Circuit Chip
PIN	Personal Identification Number
PUK	Personal Unblocking Key
QES	Qualifizierte elektronische Signatur
RFID	Radio Frequency Identification
SigG	Signaturgesetz
SIM	Subscriber Identity Module
SMS	Short Message Service
SVG	Scalable Vector Graphics
TLS	Transport Layer Security
UI	User Interface
UMTS	Universal Mobile Telecommunications System
WAP	Wireless Application Protocol
WTLS	Wireless Transport Layer Security

Literaturverzeichnis

[1] Itoi, Naomaru ; Fukuzawa, Tomoko ; Honeyman, Peter: Secure Internet Smartcards. In: *Java on Smart Cards:Programming and Security*, Springer Berlin / Heidelberg, 2001, S. 73–89

[2] Hinz, Walter: Authentication for Web Services with the Internet Smart Card. In: *ISSE 2008 Securing Electronic Business Processes*, Vieweg+Teubner, 2009, S. 357–366

[3] Bundesamt für Sicherheit in der Informationstechnik: *Advanced Security Mechanisms for Machine Readable Travel Documents*. 2009. – https://ssl.bsi.bund.de/english/publications/techguidelines/tr03110/TR-03110_v201.pdf

[4] Bundesministerium des Innern: *Einführung des elektronischen Personalausweises in Deutschland Grobkonzept - Version 2.0*. 2008. – http://www.bmi.bund.de/cae/servlet/contentblob/122648/publicationFile/9169/Grobkonzept_Personalausweis.pdf

[5] Bundesregierung Deutschland: *Entwurf eines Gesetzes über Personalausweise und den elektronischen Identitätsnachweis sowie zur Änderung weiterer Vorschriften*. 2008. – http://dip21.bundestag.de/dip21/btd/16/104/1610489.pdf

[6] Bundesregierung Deutschland: *Kabinett beschließt neuen Personalausweis mit Internetfunktion*. 2008. – http://www.bmi.bund.de/cln_095/SharedDocs/Pressemitteilungen/DE/2008/07/e_personalausweis.html

[7] Bundesamt für Sicherheit in der Informationstechnik: *Grundlagen der elektronischen Signatur*. 2006. – http://www.bsi.bund.de/esig/esig.pdf

[8] Bundesregierung Deutschland: *Gesetz über Rahmenbedingungen für elektronische Signaturen - SigG*. 2001. – http://bundesrecht.juris.de/sigg_2001/

[9] ISO/IEC: *Identification cards – Integrated circuit cards, ISO/IEC 7816*. 2004

[10] ISO/IEC: *Information technology – Digital compression and coding of continuous-tone still images, ISO/IEC 10918*. 1994

[11] ISO/IEC: *Information technology – JPEG 2000 image coding system, ISO/IEC 15444*. 2005

[12] Delac, Kresimir ; Grgic, Mislav ; Grgic, Sonja: Effects of JPEG and JPEG2000 compression on face recognition. In: *in Proceedings of ICAPR 2005, LNCS 3687*, Springer-Verlag, 2005, S. 136–145

[13] Grgic, Sonja ; Mrak, Marta ; Grgic, Mislav: Comparison of JPEG Image Coders. In: *Proc. of the 3rd International Symposium on Video Processing and Multimedia Communications, VIPromCom-2001*, 2001, S. 79–85

[14] ISO/IEC: *Identification cards – Contactless integrated circuit cards – Proximity cards, ISO/IEC 14443*. 2008

[15] Deutscher Bundestag, PuK 2 P.: *Innenausschuss macht Weg frei für neue Personalausweise*. 2008. – http://www.bundestag.de/aktuell/hib/2008/2008_344/01.html

[16] Der Rat der Europäischen Union: *Normen für Sicherheitsmerkmale und biometrische Daten in von den Mitgliedstaaten ausgestellten Pässen und Reisedokumenten*. 2004. – http://europa.eu/legislation_summaries/justice_freedom_security/free_movement_of_persons_asylum_immigration/l14154_de.htm

[17] Bishop, Matt: *Computer Security: Art and Science*. Addison Wesley Professional, 2003. – ISBN 8129701847

[18] Organization, International Civil A.: *Doc 9303, Machine Readable Travel Documents, Part 1-3*. – http://www2.icao.int/en/MRTD/Pages/Doc9393.aspx

[19] Hoepman, Jaap henk ; Hubbers, Engelbert ; Jacobs, Bart ; Oostdijk, Martijn ; Schreur, Ronny W.: Crossing borders: Security and privacy issues of the European e-passport. In: *1st IWSEC (Kyoto)*, Springer, 2006, S. 152–167

[20] Riscure: *Privacy Issue in Electronic Passport*. 2006. – http://www.riscure.com/contact/privacy-issue-in-electronic-passport.html

[21] Barker, Elaine ; Barker, William ; Burr, William ; Polk, William ; Smid, Miles: Recommendation for Key Management — Part 1: General. In: *NIST Special Publication 800-57, August 2005, National Institute of Standards and Technology. Available at http://csrc.nist.gov/publications/nistpubs/800-57/SP800-57-Part1.pdf*, 2005

[22] Bernstein, Daniel J.: Understanding brute force, 2005. – http://www.ecrypt.eu.org/stream/papersdir/036.pdf

[23] Kügler, Dennis: *Anwendungsstrategien für PACE*. 2008

[24] Büger, Matthias: Deployment of German Electronic Citizen Cards in Banking: Opportunities and Challenges. In: *ISSE 2008 Securing Electronic Business Processes*, Vieweg+Teubner Verlag, 2008

[25] Zipfel, Christian ; Daum, Henning ; Meister, Gisela: Secure E-Business applications based on the European Citizen Card. In: *ISSE 2008 Securing Electronic Business Processes*, Vieweg+Teubner Verlag, 2008

[26] Bundesregierung Deutschland: *Gesetz über das Aufspüren von Gewinnen aus schweren Straftaten - GwG*. 2008. – http://bundesrecht.juris.de/gwg_2008/

[27] Bundesministerium für Wirtschaft und Technologie: *Gesetz über das Verfahren des elektronischen Entgeltnachweises*. 2009. – http://www.bmwi.de/BMWi/Redaktion/PDF/Gesetz/elena.pdf

[28] Stickel, Eberhard: Public-Key Cryptography. In: *Encyclopedia of Information Science and Technology (IV)*, 2005, S. 2368–2372

[29] Rivest, R.L. ; Shamir, A. ; Adleman, L.: A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. In: *Communications of the ACM* 21 (1978), S. 120–126

[30] ipoque GmbH: *Internetstudie 2007*. – www.ipoque.com/userfiles/file/p2p_study_2007_abstract_de.pdf

[31] Internet Engineering Task Force: *HTTP Over TLS - RFC 2818*. 2000. – http://tools.ietf.org/html/rfc2818

[32] Menezes, Alfred J. ; Oorschot, Paul C. ; Vanstone, Scott A.: *Handbook of Applied Cryptography*. Crc Press, 1996. – ISBN 0849385237

[33] Itkis, Gene: *Forward security, adaptive cryptography: Time evolution*. 2004

[34] Diffie, Whitfield ; Hellman, Martin E.: New Directions in Cryptography. In: *IEEE Transactions on Information Theory* IT-22 (1976), Nr. 6, S. 644–654

[35] Menezes, Alfred J. ; Oorschot, Paul C. V. ; Vanstone, Scott A. ; Rivest, R. L.: *Handbook of Applied Cryptography*. 1997

[36] Wang, Yong ; Ramamurthy, Byrav: The Performance of Elliptic Curve Based Group Diffie-Hellman Protocols for Secure Group Communication over Ad Hoc Networks. In: *Communications, 2006. ICC '06. IEEE International Conference on* 5 (2006), S. 2243–2248

[37] Bundesamt für Sicherheit in der Informationstechnik: *Elliptic Curve Cryptography*. 2009. – http://www.bsi.de/literat/tr/tr03111/BSI-TR-03111.pdf

[38] Standards for Efficient Cryptography Group: *SEC1. Elliptic curve cryptography*. 2000. – http://www.secg.org/download/aid-385/sec1_final.pdf

[39] Bellovin, Steven M. ; Merritt, Michael: Encrypted Key Exchange: Password-Based Protocols Secure Against Dictionary Attacks, 1992, S. 72–84

[40] Bellovin, Steven M. ; Merritt, Michael: Augmented Encrypted Key Exchange: a Password-Based Protocol Secure Against Dictionary Attacks and Password File Compromise, ACM Press, 1993, S. 244–250

[41] Jablon, David P.: Strong Password-Only Authenticated Key Exchange. In: *ACM Computer Communications Review* 26 (1996), S. 5–26

[42] Boyko, Victor ; Mackenzie, Philip ; Patel, Sarvar: Provably secure password-authenticated key exchange using Diffie-Hellman, Springer-Verlag, 2000, S. 156–171

[43] Lucks, Stefan: Open Key Exchange: How to Defeat Dictionary Attacks Without Encrypting Public Keys. In: *Proc. of the Security Protocols Workshop, LNCS 1361*, Springer-Verlag, 1997, S. 79–90

[44] Jablon, David P.: Extended Password Key Exchange Protocols Immune to Dictionary Attack. In: *Proc. of WET-ICE '97*, 1997, S. 248–255

[45] Boyd, Colin ; Mathuria, Anish: *Protocols for Authentication and Key Establishment*. Springer, 2003. – ISBN 3540431071

[46] ISO/IEC: *Information technology – Security techniques – Entity authentication – Part 2: Mechanisms using symmetric encipherment algorithms, ISO/IEC 9798*. 1999

[47] MacKenzie, Philip: *On the Security of the SPEKE Password-Authenticated Key Exchange Protocol*. Cryptology ePrint Archive, Report 2001/057, 2001. – http://eprint.iacr.org/

[48] J-S. Coron, A. G. ; Paillier, P.: Password Authenticated Secure Channel v5 (PASC5), Cryptography & Innovation, Gemalto Security Labs, 2009

[49] Ullmann, Markus ; Kügler, Dennis ; Neumann, Heike ; Stappert, Sebastian ; Vögeler, Matthias: Password Authenticated Key Agreement for Contactless Smart Cards. In: *Communications of the ACM* (2008)

[50] Needham, Roger M. ; Schroeder, Michael D.: Using encryption for authentication in large networks of computers. In: *Commun. ACM* 21 (1978), Nr. 12, S. 993–999. – ISSN 0001–0782

[51] Lowe, Gavin: An attack on the Needham-Schroeder public-key authentication protocol. In: *Information Processing Letters* 56 (1995), S. 131–133

[52] Bender, Jens ; Fischlin, Marc ; Kügler, Dennis: Security Analysis of the PACE Key-Agreement Protocol. In: *12th International Information Security Conference (ISC 2009)* (2009)

[53] Burkart, Günter: *Handymania - Wie das Mobiltelefon unser Leben verändert hat.* Campus Verlag, 2007. – ISBN 3593383519

[54] ISO/IEC: *Information processing systems – Vocabulary, ISO/IEC 2382*. 1987

[55] Canalys Ltd.: *Global smart phone shipments rise 28%*. 2008. – http://www.canalys.com/pr/2008/r2008112.pdf

[56] Heiner, Andreas P. ; Asokan, N.: Secure software installation in a mobile environment. In: *SOUPS '07: Proceedings of the 3rd symposium on Usable privacy and security*. New York, NY, USA : ACM, 2007. – ISBN 978–1–59593–801–5, S. 155–156

[57] WAP Forum: *Wireless Transport Layer Security Specification*. 2001

[58] Kettula, Arto: Security Comparison of Mobile OSes, 2000. – www.tml.tkk.fi/Opinnot/Tik-110.501/2000/papers/kettula.pdf

[59] Singelée, Dave ; Preneel, Bart: Security Overview of Bluetooth. In: *COSIC Internal Report* (2004)

[60] Mahmoud, Jaap H. ; Naghshineh, Mahmoud ; Inouye, Jon ; Joeressen, Olaf J. ; Allen, Warren: Bluetooth: Vision, Goals, and Architecture. In: *ACM Mobile Computing and Communications Review* 2 (1998), S. 38–45

[61] August, Praveen Y.: *A Survey on Security Issues in Wireless Networks*

[62] Katoen, J-P. ; Schoenmakers, B.: A UMTS Network Architecture. In: *Proceedings 2nd Workshop on Algorithms and Parallel VLSI Architectures*, 1992, S. 79–85

[63] Broll, Gregor ; Hamard, John ; Paolucci, Massimo ; Haarländer, Markus ; Siorpaes, Sven ; Rukzio, Enrico ; Schmidt, Albrecht ; Wiesner, Kevin: *Mobile Interaction with Web Services through Associated Real World Objects*. 2007

[64] ABI Research: *Near Field Communication Semiconductors*. 2007. – http://www.abiresearch.com/research/1001509-Near_Field_Communication_Semiconductors

[65] Falke, Oliver ; Rukzio, Enrico ; Dietz, Ulrich ; Holleis, Paul ; Schmidt., Albrecht: Mobile Services for Near Field Communicatio / Ludwig-Maximilians-Universität München. 2007 (LMU-MI-2007-1). – Forschungsbericht. – ISSN 1862–5207

[66] ECMA International: *Standard ECMA-340 – Near Field Communication Interface and Protocol (NFCIP-1)*

[67] ISO/IEC: *Information technology – Telecommunications and information exchange between systems – Near Field Communication – Interface and Protocol (NFCIP-1), ISO/IEC 18092*. 2004

[68] ECMA International: *Standard ECMA-352 – Near Field Communication Interface and Protocol -2 (NFCIP-2)*

[69] ISO/IEC: *Information technology – Telecommunications and information exchange between systems – Near Field Communication Interface and Protocol -2 (NFCIP-2), ISO/IEC 21481*. 2005

[70] ISO/IEC: *Identification cards – Contactless integrated circuit cards, Vicinity cards, ISO/IEC 15693*. 2004

[71] Sarma, Implications S. ; Sarma, Sanjay E. ; Weis, Stephen A. ; Engels, Daniel W.: RFID Systems and Security and Privacy. In: *In Workshop on Cryptographic Hardware and Embedded Systems*, Springer, 2002, S. 454–470

[72] Peris-lopez, Pedro ; Hern, Julio C. ; Estevez-tapiador, Juan M. ; Ribagorda, Arturo: RFID systems: A survey on security threats and proposed solutions. In: *in 11th IFIP International Conference on Personal Wireless Communications – PWC06, ser. Lecture Notes in Computer Science*, Springer-Verlag, 2006, S. 159–170

[73] E. Haselsteiner, K. B.: Security in Near Field Communication (NFC), 2006. – http://events.iaik.tugraz.at/RFIDSec06/Program/papers/002%20-%20Security%20in%20NFC.pdf

[74] Teodora Kostic: *Near Field Communication*. 2009. – http://lasecwww.epfl.ch/courses/sp09/NFC.pdf

[75] Mulliner, Collin: Vulnerability Analysis and Attacks on NFC-enabled Mobile Phones, 2009. – http://mulliner.org/collin/academic/publications/vulnanalysisattacksnfcmobilephones_mulliner_2009.pdf

[76] Kügler, Dennis: Man in the Middle Attacks on Bluetooth. In: *Financial Cryptography*, Springer Berlin / Heidelberg, 2004, S. 149–161

[77] Meyer, Ulrike ; Wetzel, Susanne: A Man-in-the-Middle Attack on UMTS. In: *in Proceedings of the 2004 ACM Workshop on Wireless Security*, ACM Press, 2004, S. 90–97

[78] Corporation, Nokia: *Contactless Communication API JSR 257*. 2009. – http://jcp.org/aboutJava/communityprocess/mrel/jsr257/index.html

[79] Gamma, Erich ; Helm, Richard ; Johnson, Ralph E.: *Design Patterns. Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1995. – ISBN 0201633612

[80] Rumbaugh, James ; Jacobson, Ivar ; Booch, Grady: *The Unified Modeling Language Reference Manual*. Addison-Wesley, 1999. – ISBN 0321245628

[81] Grand, Mark: *Patterns in Java: A Catalog of Reusable Design Patterns Illustrated with UML*. Wiley and Sons, 2002. – ISBN 0471227293

[82] Meister, Gisela ; Hühnlein, Detlef ; Eichholz, Jan ; Araújo, Roberto: eVoting with the European Citizen Card. In: *BIOSIG*, 2008, S. 67–78. – http://www.ecsec.de/pub/ECC-voting.pdf

[83] Balzer, Mara: *Mobiles Bezahlen*. 2005. – http://www.hcilab.org/events/mobileinteraction/reports/10_MobilesBezahlen_MaraBalzer.pdf

[84] Lenz, Harald: *M-Payment, Zahlungsmethoden im M-Commerce*. 2004. – http://michael.hahsler.net/stud/done/lenz/Diplomarbeit_Lenz_M-Payment2004.pdf

[85] Müller, Nils ; Reiprich, Thorsten: *J2ME Polish*. 2008. – http://www.mi.fh-wiesbaden.de/~barth/mobile/ws0708/J2MEPolish.pdf

[86] Bundesamt für Sicherheit in der Informationstechnik: *eCard-API-Framework, BSI TR-03112*. 2008. – http://www.bsi.bund.de/literat/tr/tr03112/index.htm

[87] Hühnlein, Detlef ; Bach, Manuel ; Oberweis, Rainer: Das eCard-API-Framework. In: *10. Deutscher IT-Sicherheitskongress des BSI* (2007)

[88] Walloschke, Thomas: Infrastructures and Middleware for the Application of eID Cards in eGovernment. In: *ISSE 2008 Securing Electronic Business Processes*, Vieweg+Teubner, 2009, S. 406–417

[89] KNUDSEN, Jonathan: *Parsing XML in J2ME*. 2002. – http://developers.sun.com/mobility/midp/articles/parsingxml/

MobilePACE
Bachelor-Thesis von Moritz Horsch
Technische Universität Darmstadt

86 Pages | 22772 Words

This document was created with LaTeX.