Fachbereich Informatik

Studiengang Angewandte Informatik

Institut für Internet-Sicherheit, if(is)

Master Thesis

Web-Authentisierung mit dem elektronischen Personalausweis (nPA)

Juli 2008

Christian J. Dietrich

Betreuer

Prof. Dr. Norbert Pohlmann

_____

Datum und Unterschrift

Prof. Dr. Andreas Cramer

_____

Datum und Unterschrift

Eidesstattliche Erklärung

Hiermit versichere ich, die Arbeit selbständig angefertigt und keine anderen als die angegebenen und bei Zitaten kenntlich gemachten Quellen und Hilfsmittel benutzt zu haben.

Gelsenkirchen, den 07.07.2008, _____

Inhaltsverzeichnis

1 Einleitung

Seit ihrer Gründung im Jahre 1944 beschäftigt sich die International Civil Aviation Organization (kurz ICAO, dt. Internationale Organisation für die zivile Luftfahrt), eine Sonderorganisation der Vereinten Nationen, mit der Normung im Luftfahrtbereich (vgl. [ICAO 2006]). Heutzutage sind rund 190 Staaten Mitglied der ICAO. Im Jahr 1998 beschloss die ICAO die Intensivierung der Sicherheitsmechanismen von Reisedokumenten. Dies umfasst unter anderem die Integration eines elektronischen Speichers in Reisedokumente. Dieser Prozess wurde insbesondere im Jahr 2001 nach den Anschlägen des 11. Septembers bekräftigt (siehe [BMI 2004 Sep11]). Darüber hinaus verabschiedete die ICAO im Rahmen der sog. Blueprint-Veröffentlichungen grundlegende Sicherheitsmerkmale zukünftiger Reisedokumente (vgl. [ICAO-BIOM-MRTD03]). Dies umfasst beispielsweise die Verwendung von Radio Frequency, also Funktechnik, zum Auslesen von elektronisch gespeicherten Daten, die Speicherung eines Lichtbilds und weiterer biometrischer Merkmale, die Verwendung einer einheitlichen logischen Datenstruktur sowie eine Public Key Infrastruktur für kryptographische Zwecke (vgl. [ICAO-BIOM-MRTD03]).

Auf europäischer Ebene wurde das Thema im Jahr 2004 aufgegriffen und im Dezember 2004 in der EU-Verordnung Nr. 2252/2004 über Normen für Sicherheitsmerkmale und biometrische Daten in von den Mitgliedstaaten ausgestellten Pässen und Reisedokumenten manifestiert (vgl. [EU/EG Nr.2252/2004]). Dies führte letztlich dazu, dass die deutsche Bundesregierung im Jahr 2005 die Einführung des elektronischen Reisepasses beschloss. Die gesetzliche Grundlage auf nationaler Ebene bildet die entsprechende Änderung des deutschen Passgesetzes, die im Mai 2007 wirksam wurde. Seit November 2007 werden elektronische Reisepässe in Deutschland mit Fingerabdrücken und einem Lichtbild in elektronisch gespeicherter Form ausgegeben.

1.1 Vom elektronischen Reisepass zum elektronischen Personalausweis

Die Einführung des elektronischen Reisepasses ist damit abgeschlossen. Sie war der erste Schritt in Richtung der Umstellung von herkömmlichen auf maschinenlesbare und elektronische Ausweisdokumente, sog. machine readable travel documents (MRTD). Die Bundesregierung plant ab 2009 den elektronischen Personalausweis in Deutschland einzuführen (vgl. [Reisen 2008]). Die Einführung des elektronischen Personalausweises ist ein weiterer Teil des vom Bundeskabinett im September 2006 beschlossenen Programms zur Modernisierung der Bundesverwaltung. Im Gegensatz zum elektronischen Reisepass gibt es derzeit jedoch auf internationaler oder europäischer Ebene noch keine gesetzlichen Rahmenbedingungen. In Deutschland ist nach Aussage des zuständigen Ministerialrats im Bundesinnenministerium davon auszugehen, dass auf nationaler Ebene das Gesetzgebungsverfahren für den elektronischen Personalausweis im Laufe des Jahres 2008 abgeschlossen sein wird (vgl. [Reisen 2008]). Warum ist die Einführung eines elektronischen Personalausweises insbesondere vor dem Hintergrund des bereits eingeführten elektronischen Reisepasses sinnvoll?

Aufgrund der Ausweispflicht benötigen wir in Deutschland den Personalausweis in erster Linie, um unsere Identität für Dritte verifizierbar zu machen. Dies ist beispielsweise beim Check-In im Hotel oder bei der Beantragung von Vorgängen im behördlichen Bereich der Fall. Eine nachvollziehbare Verifikation der Identität konnte bisher lediglich offline stattfinden, z.B. durch Verfahren wie PostIdent der Deutschen Post ([DP PostIdent]) oder manuelles Vorlegen und Verifizieren des Personalausweises. Selbst Prozesse, die als solche im Grunde ausschließlich online ablaufen könnten, erfordern aufgrund der Identitätsprüfung einen Offline-Interaktionsschritt. Darüber hinaus findet in vielen Bereichen ein datenschutzrechtlich fragwürdiger Umgang mit sensiblen Daten statt. Die Erforderlichkeit für die Anfertigung einer Kopie eines Personalausweises – wie mitunter beim Hotel-Check-In der Fall – ist nicht zweifelsfrei geklärt. Reicht es nicht aus, die Identität der eincheckenden Person zu klären ohne eine Kopie des Ausweises anzufertigen?
Das Aufbewahren der Kopien von Ausweisen führt zwangsläufig zu einem

Sammelsurium an personenbezogenen Daten. Dies ist aus datenschutzrechtlicher Perspektive fraglich und aus Sicht der Bürger unerwünscht. Ferner hat der Ausweisinhaber keine Kontrolle mehr über die Weiterverwendung seiner Daten.
Ein elektronischer Personalausweis (kurz ePA oder nPA) kann helfen, diese Missstände zu beseitigen. Darüber hinaus soll der ePA laut [Roßnagel et. al 2008]

> „[...] den bisherigen Personalausweis so fortentwickeln, dass er seine Grundaufgabe des Identitätsnachweises in Zukunft auch in elektronischen Rechts- und Geschäftsprozessen erfüllen kann."

Ein weiterer Vorteil des elektronischen Personalausweises ist die Verwendung als Passersatz im Schengenraum. Die Ausweispflicht hat eine weite Verbreitung des Personalausweises in Deutschland zur Folge. Der Reisepass hingegen ist nicht derart weit verbreitet.[1]

1.2 Ziele dieser Arbeit

Diese Arbeit ist im Auftrag des Bundesministerium des Innern (BMI) entstanden. Zunächst soll diese Arbeit die Notwendigkeit und die Vergleichbarkeit des Sicherheitsniveaus zwischen dem physikalischen und dem elektronischen Ausweisdokument aufzeigen. Darüber hinaus soll eine Implementierung der Authentisierungsfunktion des ePA für die Anwendung über das Internet – die erste ihrer Art – ihre Machbarkeit demonstrieren. Im Rahmen der Arbeit werden also sowohl die notwendigen Komponenten aller teilnehmenden Parteien vorgestellt, die für eine Realisierung benötigt werden als auch tatsächliche Implementierungen dieser Komponenten beschrieben und gefertigt. Der Ablauf der Online-Authentisierung soll zusätzlich aus der Perspektive des Anwenders dokumentiert werden. Hierbei sind wichtige Schritte detailliert zu beschreiben.

1 Um geringfügig vergleichbare Zahlen zu nennen: Im November 2007 befanden sich etwa 62 Millionen *Personalausweise* von bundesdeutschen Bürgern im Umlauf ([BReg 2007]). Hierbei sollte beachtet werden, dass es sich um eine Momentaufnahme handelt. Laut [Beel und Gipp 2005] gibt die Bundesdruckerei an, dass sie seit seiner Einführung 1988 mehr als 65 Millionen *Reisepässe* produziert habe.

Ein weiterer Kernaspekt der Arbeit ist ferner die Betrachtung von Schwachstellen und Sicherheitslücken der Online-Authentisierungsfunktion des ePA. Nachdem die Beispiel-Implementierung modelliert und gefertigt wurde, sollen mögliche Angriffe und ihre Auswirkungen auf die Sicherheit des Prozesses durchgeführt werden. Sofern eine Schwachstelle aufgedeckt wird, sollen Vorschläge erarbeitet werden, die zur Behebung des Sicherheitsproblems führen.

2 nPA/ePA – der elektronische Personalausweis

Der elektronische Personalausweis wird in seiner Funktionalität dem elektronischen Reisepass ähneln. Aller Voraussicht nach wird es sich – wie beim Reisepass – im Kern um einen kontaktlosen Chip (RF-Chip) handeln (vgl. [Kuegler und Naumann 2007]). Er kommuniziert über Funk (RF) nach ISO 14443 ([ISO/IEC 14443]) mit einem RF-Lesegerät. Vor dem Hintergrund der 10-jährigen Gültigkeit des Personalausweises wurde auch hier eine Funkschnittstelle gewählt, da diese über eine geringere Materialabnutzung im Vergleich zu kontaktbasierten Verfahren verfügt. Der Chip kann diverse Daten speichern und besitzt einen kryptografischen Koprozessor. Möglicherweise wird der ePA die Größe einer Kreditkarte haben (Formfaktor ID-1 nach [ISO/IEC 7810]) und wie bisher ein sichtbares Lichtbild sowie aufgedruckte personenbezogene Angaben und Sicherheitsmerkmale enthalten.

Abbildung 1: Entwurf des elektronischen Personalausweises nach [BMI Grobkonzept ePA]

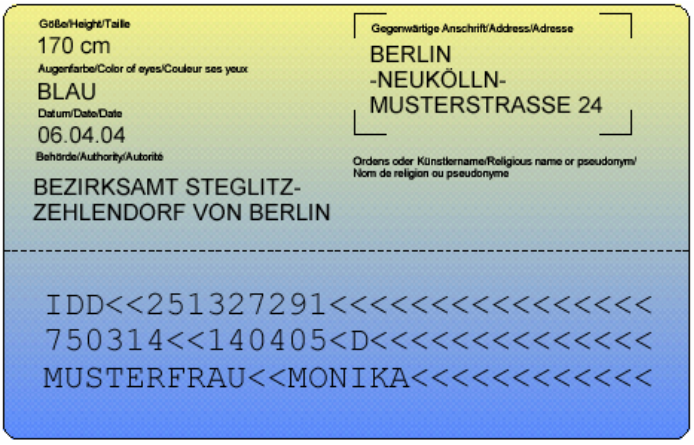


Abbildung 2: Entwurf der Rückseite des elektronischen Personalausweises nach [BMI Grobkonzept ePA]

Neben den schon heute verfügbaren physikalischen Funktionen zur Verifizierung der Identität der tatsächlich zur Verfügung stehenden Person, hat der elektronische Personalausweis die zusätzliche Möglichkeit der elektronischen Authentisierung und optional auch die Funktion der qualifizierten Signatur. Diese zusätzlichen Funktionen sollen auch helfen, die Vertrauenswürdigkeit der elektronischen Abläufe in unserer vernetzten Wissens- und Informations-gesellschaft zu erhöhen. Die IT-basierten Abläufe können möglicherweise mit Hilfe des elektronischen Personalausweises zukünftig ein höheres Sicherheitslevel aufweisen.

Der elektronische Reisepass hat mit der Einbindung des RF-Chips seine Bauform nur geringfügig verändert. Abbildung 3 zeigt den schematischen Aufbau des Reisepasses. In den Buchdeckel wurde der RF-Chip sowie die dazugehörige Antenne für die Funkkommunikation eingearbeitet. Ein Symbol auf dem Passdeckel kennzeichnet den elektronischen Reisepass und grenzt ihn optisch von den älteren, nicht-elektronischen ab.

Abbildung 3: Schematische Darstellung des Aufbaus des deutschen elektronischen Reisepasses.
Quelle: Bundesministerium des Innern

2.1 Physikalische Sicherheitsmerkmale des bisherigen Personalausweises und Reisepasses

Der bisherige nicht-elektronische Personalausweis sowie der Reisepass besitzen eine Reihe an physikalischen Sicherheitsmerkmalen, anhand derer die Authentizität des Ausweisdokuments überprüft werden kann. Ein Großteil dieser Sicherheitsmerkmale wurde nach dem 11. September 2001 völlig neu entwickelt. Im Rahmen der Identitätsfeststellung nimmt die Überprüfung der physikalischen Sicherheitsmerkmale einen wichtigen Stellenwert ein, da nur ein echtes Ausweisdokument eine positive Identitätsfeststellung zur Folge haben darf. Im

Folgenden sind die physikalischen Sicherheitsmerkmale am Beispiel des aktuellen nicht-elektronischen Personalausweises (Stand Juni 2008) aufgeführt.

Abbildung 4: Normalansicht des deutschen nicht-elektronischen Personalausweises. Quelle: Bundesdruckerei

Abbildung 5: Ansicht des deutschen nicht-elektronischen Personalausweises unter flachem Betrachtungswinkel. Quelle: Bundesdruckerei

Abbildungen 4 und 5 zeigen die folgenden, physikalischen Sicherheitsmerkmale des aktuellen, nicht-elektronischen Personalausweises (Stand Juni 2008):

1. Das Portraitbild in Schattendarstellung zeigt in holografischer Form eine Hell-Dunkel-Wiedergabe des Bildinhalts und enthält am linken Bildrand vier Bundesadlermotive.
2. Der rötliche Bundesadler in Pseudo-3D-Darstellung ist nur unter einem bestimmten Betrachtungswinkel erkennbar.
3. Im oberen Teil des Portraitfotos sind sog. kinematische Bewegungsstrukturen eingearbeitet. Diese stellen einen von 12 Sternen umgebenen Bundesadler dar. Wenn der Ausweis seitlich geneigt wird,

sind je nach Winkel entweder ein Sechseckmotiv oder der Großbuchstabe D zu erkennen.

4. Der linke Rand des herkömmlichen Portraitfotos enthält in geschwungener Mikroschrift den Text „BUNDESREPUBLIK DEUTSCHLAND“. Darüber hinaus sind mehrere parallel verlaufende Mikroschriftzeilen, ebenfalls mit dem Text „BUNDESREPUBLIK DEUTSCHLAND“ enthalten.
5. Durch Auf- und Abkippen des Ausweises wird der holografische Bundesadler in umgekehrten Kontrastverhältnissen dargestellt.
6. Die beiden maschinenlesbaren Zeilen des Ausweises werden – leicht nach oben versetzt – zusätzlich holografisch dargestellt.
7. Ein weiteres holografisches Motiv ist im mittleren, unteren Bildbereich des Portraitfotos enthalten.
8. Durch Oberflächenprägung sind die Schriftzüge „BUNDESREPUBLIK DEUTSCHLAND“ sowie „PERSONALAUSWEIS“ auch erfühlbar.
9. Mehrfarbige Schutzmuster aus feinen miteinander verschlungenen Linien, sog. Guillochen, sollen Reproduktionen verhindern. Üblicherweise können Farbkopierer diese Linien nur in gerasteter Form wiedergeben.
10. Vor- und Nachname werden am rechten Portraitfotorand in die Oberfläche eingebrannt.
11. Bei durchscheinendem Licht zeigt sich ein mehrstufiges Wasserzeichen in Form von über die Fläche verteilten Bundesadlern.

Zum Vergleich zeigen die folgenden beiden Abbildungen diese physikalischen Sicherheitsmerkmale beim deutschen Reisepass.

Abbildung 6: Normalansicht des deutschen Reisepasses. Quelle: Bundesdruckerei

Abbildung 7: Ansicht des deutschen Reisepasses unter flachem Betrachtungswinkel. Quelle: Bundesdruckerei

Deutsche Ausweisdokumente weisen also eine Vielzahl an physikalischen Sicherheitsmerkmalen auf. Diese können im Bedarfsfall helfen, die Authentizität des Ausweisdokuments zu klären.
Problematisch ist in diesem Zusammenhang fehlendes Wissen der kontrollierenden Einheit. Sofern die Authentizität des Ausweisdokuments nicht erfolgreich geprüft werden kann – beispielsweise wenn die Sicherheitsmerkmale dem Kontrolleur nicht bekannt sind oder die Gegebenheiten der Umgebung verhindern, dass sie geprüft werden können wie etwa durch mangelndes Licht – kann auch keine Aussage zu den auf dem Ausweisdokument vermerkten Daten gemacht werden. Je komplexer die physikalischen Sicherheitsmerkmale, umso aufwändiger und schwieriger wird die Kontrolle.

2.1.1 Fälschungsdelikte

Die deutschen Ausweisdokumente Personalausweis und Reisepass gelten weltweit als eine der sichersten und nur sehr schwer fälschbaren Dokumente (vgl. [BMI Fälschungssicherheit]). Dass trotzdem immer wieder Fälschungen auftauchen, zeigt die Antwort der Bundesregierung auf die Kleine Anfrage vom November 2007, in der zum ersten Mal Statistiken zu Ausweisfälschungen von Personalausweisen veröffentlicht werden. Im November 2007 befanden sich rund 62 Millionen Personalausweise im Umlauf (vgl. [BReg 2007]). Auf die Frage nach Fälschungen antwortete die Bundesregierung:

> „Für den Zeitraum vom 1.Januar bis einschließlich 30.September 2007 sind insgesamt 495 Urkundendelikte im Zusammenhang mit der Nutzung **deutscher Personalausweise** an den Grenzen oder im Inland festgestellt worden und in der Statistik registriert. Dabei handelt es sich in 88 Fällen um Totalfälschungen sowie in 128 Fällen um Verfälschungen von deutschen Personalausweisen. Die hauptsächlichen Verfälschungsmerkmale waren Änderung der Ausstellungsdaten (49 Fälle) sowie das Überschreiben/Anfügen von Daten (47 Fälle). Darüber hinaus entfielen 259 Feststellungen auf Ausweismissbräuche.“ ([BReg 2007], S. 2)

Darüber hinaus sind keine Fälle bekannt, in denen deutsche Personalausweise eine Rolle im Zusammenhang mit durchgeführten oder geplanten und aufgedeckten oder verhinderten terroristischen Anschlägen seit 2000 gespielt haben.

Ähnlich sieht die Situation bei Fälschungsdelikten mit deutschen Reisepässen aus. Der ehemalige Bundesinnenminister Otto Schily wies in einem Interview mit dem SPIEGEL darauf hin, dass die Fälschung von Reisepässen Realität ist (vgl. [BMI 2004 Sep11], S.40f). Im Jahr 2002 untersuchte der Bundesgrenzschutz 7700 Reisepässe (vgl. [BReg 2005]), von denen 383 Dokumente Totalfälschungen waren, dies entspricht rund 5% der untersuchten Reisepässe. Darüber hinaus handelte es sich bei 1480 Fällen um inhaltlich veränderte Originalpässe (~19%) und in 91 Fällen um entwendete Blankovordrucke (~1%). Deutsche Pässe waren in nur 35 Fällen betroffen. Zu beachten ist hierbei jedoch, dass sich ein Großteil der 35 deutschen Fälschungen auf vorläufige Pässe bezog. Schwerpunktmäßig waren die Pässe der Länder Italien, Frankreich und Spanien betroffen. Nach Aussage von Otto Schily sind zu dieser Anzahl eine Menge weiterer Fälle zu zählen, die von Strafverfolgungsbehörden aufgedeckt wurden und nicht vom Grenzschutz. Zu letzterem sind allerdings keine Zahlen veröffentlicht worden.

2.1.2 Übertragung der Sicherheitsmerkmale auf die elektronische Funktionalität des Ausweises

Insbesondere im Kontext digitaler Ausweisdokumente ist offensichtlich, dass die elektronische (Zusatz-)Funktionalität der Ausweise ein vergleichbares Maß an Sicherheit in Relation zur nicht-elektronischen Anwendung aufweisen sollte. Die

gewählten elektronischen Sicherheitsmerkmale sind Biometrie und die Verwendung eines funkbasierten Smartcard-Chips. Diese Merkmale sollen insbesondere im Fall von hoheitlichen Anwendungen die Effektivität von Grenzkontrollen und hier insbesondere die Echtheitsprüfung und die Integrität des Dokuments erleichtern.
Zum Zugriffsschutz wurden von der ICAO die folgenden beiden Mechanismen vorgeschlagen:

- Basic Access Control (BAC)
- Extended Access Control (EAC)

2.1.2.1 Basic Access Control

Basic Access Control (kurz BAC) ist ein einfacher Zugriffsschutz. Mit Hilfe von BAC wird der Zugriff auf weniger sensitive, elektronisch gespeicherte Daten des Ausweisdokuments durch einen Schlüssel beschränkt, der sich aus der sog. maschinenlesbaren Zone des Dokuments ableitet. Unter der maschinenlesbaren Zone (kurz: MRZ, engl. machine readable zone) versteht man den Teil des Dokuments, der mit Hilfe von Optical Character Recognition (OCR) gelesen werden kann. Die MRZ umfasst dabei häufig den Namen, die Seriennummer des Dokuments, die Nationalität, das Geburtsdatum und das Geschlecht (vgl. [ICAO-MRTD-LDS]). Den entsprechenden Ausschnitt aus der ICAO-Spezifikation für die erforderlichen Daten der maschinenlesbaren Zone des Reisepasses zeigt Abbildung 8 in dem Bereich betitelt mit **Detail(s) Recorded in MRZ**.

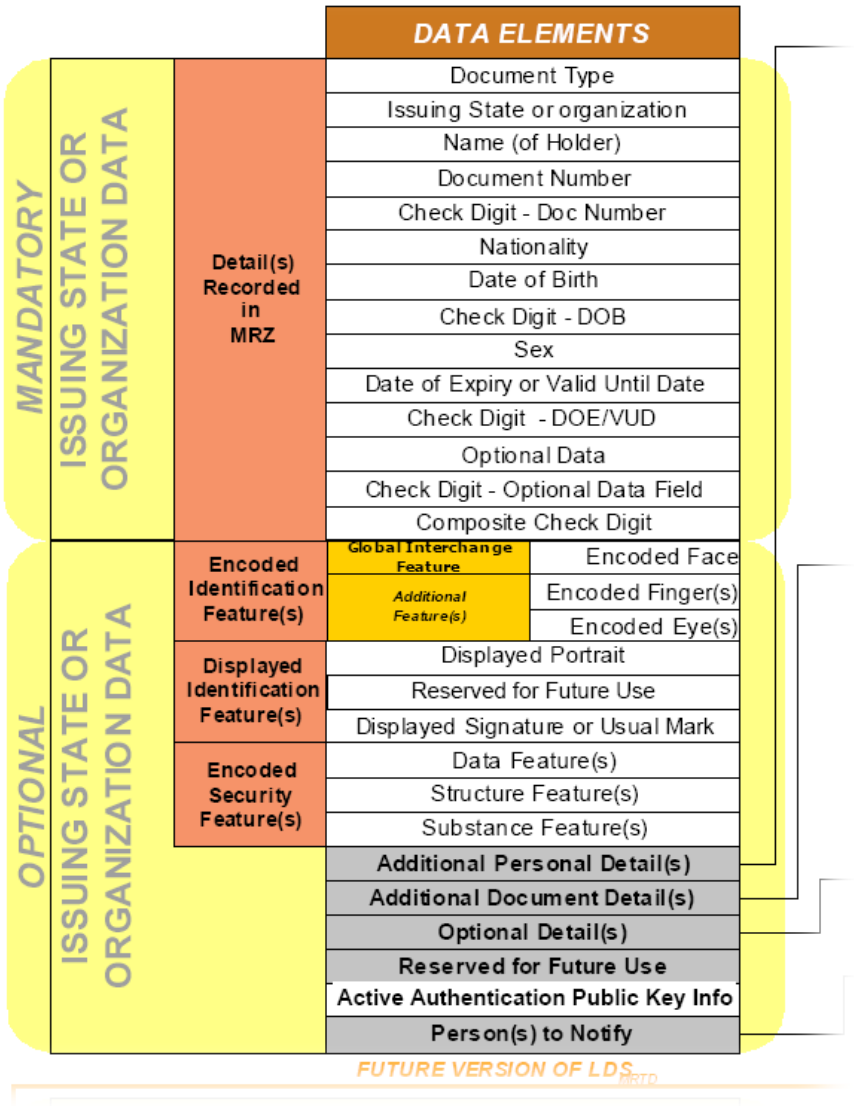


Abbildung 8: Ausschnitt aus der ICAO-Spezifikation für die Datenspeicherung der maschinenlesbaren Zone (MRZ). DOB=Date Of Birth, DOE=Date Of Expiry, VUD=Valid Until Date. Quelle: ICAO, Figure V.1 aus [ICAO-MRTD-LDS]

Bevor die Daten des Ausweisdokuments elektronisch übermittelt werden können, muss aus den Daten der MRZ ein Schlüssel gebildet werden. Mit Hilfe dieses Schlüssels erhält das Lesegerät Zugriff. Der entscheidende Aspekt bei diesem Verfahren ist, dass ein Lesen der MRZ mittels OCR nur dann möglich ist, wenn das Dokument auf einen Scanner gelegt wird, damit dieser die MRZ erfassen kann. Der elektronische Zugriff basiert damit – wie der physikalische Zugriff – auf der optischen Einsichtnahme in das Ausweisdokument. Die Schlüsselgenerierung erfolgt ausschließlich aus prüfsummengestützten Teilen der MRZ, üblicherweise sind dies:

- die Seriennummer des Dokuments (Document Number)
- das Geburtsdatum (Date of Birth) und
- das Ablaufdatum (Date of Expiry)

Dieser Schlüsselraum beläuft sich auf theoretisch maximal 56 Bits (vgl. [ICAO-PKI-MRTD-ICC]). In der Realität lässt sich bei einem gezielten Angriff auf die Daten eines spezifischen Dokuments der Schlüsselraum der prüfsummengestützten Teile der MRZ signifikant einschränken. Das folgende Kriterium zeigt den Erfolg des Angriffs anhand von Beispielen auf:
Die Seriennummer von Reisepässen wird in der Regel mit einer Entropie von 10^9 vergeben. In Deutschland werden diese jedoch nicht zufällig vergeben, sondern es fließt die Kennzahl der zuständigen Passbehörde (maximal 6500 in Deutschland) ein. Bei bekannter Behördenkennzahl und unbekannter fünfstelliger Seriennummer reduziert sich laut [Beel und Gipp 2005] die Anzahl an Möglichkeiten von 10^9 auf 10^5.
Dieser Umstand hat bei niederländischen Pässen bereits erfolgreiche Attacken ermöglicht (vgl. [Heise HollPass]):

> „Der [...] Ansatz fußt auf der Tatsache, dass niederländische Behörden eine fortlaufende Nummerierung der Reisepässe vornehmen. Da die Zahl der monatlich ausgegebenen Pässe annähernd konstant ist, besteht darüber hinaus ein einfacher linearer Zusammenhang zwischen dem Ausgabedatum und der Passnummer. Dadurch reduziert sich die effektive Verschlüsselungsstärke der abgehörten Datenübertragung auf lediglich rund 35 Bit. Selbst ein durchschnittlicher PC benötigt nur wenige Stunden, um unter Berücksichtigung dieser Zusammenhänge mit einer so genannten Brute-Force-Attacke alle 2^{35} möglichen Schlüssel durchzuprobieren.“ ([Heise HollPass])

Auch belgische Reisepässe sind nach [Heise BelgPass] offenbar von einem ähnlichen Phänomen betroffen. Die Art und Weise der Vergabe der Seriennummer hat also signifikanten Einfluss auf das Sicherheitsniveau des Zugriffsverfahren Basic Access Control. Dies gilt ebenfalls für alle weiteren relevanten Teile für die Schlüsselgenerierung.

Die erfolgreichen Angriffe auf durch BAC geschützte Dokumente zeigen mögliche Schwächen des Basic Access Control auf. Wichtig bei der Bewertung

der Sicherheit von BAC ist jedoch, dass BAC lediglich einen vergleichbaren Schutz zur optischen Einblicknahme in das Ausweisdokument herstellen soll. Sobald optisch Zugriff auf die MRZ besteht, kann der Schlüssel zum elektronischen Auslesen abgeleitet werden.[2] [Kuegler und Naumann 2007] zeigen für den deutschen Reisepass die Dauer eines Brute-Force-Angriffs bis zum Erfolg bei unterschiedlichen Einschränkungen des Suchraums. Im Worst Case, d.h. bei Einschränkung des Suchraums auf 2^{20} mögliche Schlüssel, berechnen sie die maximale Dauer jedoch immer noch auf 12 Tage.[3]

Die erfolgreichen Angriffe zeigen jedoch zumindest den Grund, warum BAC für den Schutz sensitiver Informationen, wie biometrische Daten, nicht ausreicht. Hier würde bei einem erfolgreichen Angriff ein enormer Informationsgewinn auf Seite des Angreifers stattfinden.

2.1.2.2 Extended Access Control

Extended Access Control (EAC, dt. erweiterter Zugriffsschutz) ist im Kontext von maschinenlesbaren Reisedokumenten die Zugriffsmethode auf sensitive Daten, wie beispielsweise die biometrischen Informationen des Fingerabdrucks. Ziel von EAC ist die Sicherstellung, dass lediglich autorisierte Lesegeräte auf personenbezogene und biometrische Daten zugreifen können. Grundsätzlich finden hier Verfahren der digitalen Signatur sowie asymmetrischer Verschlüsselung Anwendung. Algorithmen für die Verwendung mit EAC sind beispielsweise Diffie-Hellman (DH), DSA oder RSA bzw. die jeweiligen Varianten für elliptische Kurven-Kryptographie ECDH und ECDSA.
Eine elektronische Signatur garantiert die Unversehrtheit der gespeicherten Daten auf dem Chip (vgl. [BSI-TR-03110]) (Authentizität). Diese Signatur kann von dem Lesegerät verifiziert werden. Die dahinter stehende Public Key Infrastruktur ist eine globale PKI, die die Mitglieder der ICAO zur Verfügung stellen müssen. Im Falle des elektronischen Reisepasses ist in Deutschland der relevante Teil dieser PKI bereits etabliert. Die Wurzelinstanz (Root CA) für in Deutschland ausgestellte Reisepässe liegt beim Bundesamt für Sicherheit in der Informationstechnik (BSI) (vgl. [Kuegler und Naumann 2007]).
Neben der passiven Sicherheit, die durch die elektronische Signatur umgesetzt

2 Möglicherweise ist ein Angriff, der zunächst auf optische Art und Weise (beispielsweise durch Überwachungskameras, die den geöffneten Pass „abfilmen“) oder auf anderem Wege die Daten der MRZ ausliest, effizienter als ein Brute-Force-Angriff, selbst mit eingeschränktem Schlüsselraum.
3 Während dieses Zeitraums muss permanent optimaler Funkkontakt zu dem Chip bestehen.

wird, bietet Extended Access Control einen Mechanismus, der die Authentizität des Chips selbst – komplementär zu den von ihm gespeicherten Daten – überprüft. Diese Funktion wird als Chip Authentication (dt. Chip Authentisierung) bezeichnet (vgl. [BSI-TR-03110]). Hierbei ist ein individueller privater Schlüssel in einem nicht-auslesbaren Bereich des Chips hinterlegt – ähnlich der Funktionalität einer SmartCard. Die Chip Authentication bietet gegenüber einer herkömmlichen aktiven Authentisierung z.B. durch ein Challenge-Response-Verfahren, den Vorteil, dass hier keine beliebigen Daten zum Signieren „untergeschoben" werden können. [Kuegler und Naumann 2007] beschreiben dies wie folgt:

> „Bei der Chip Authentisierung wird der Nachweis über die Kenntnis des privaten Schlüssels indirekt über den Aufbau eines stark verschlüsselten und integritätsgesicherten Kanals erbracht. Der dabei ausgehandelte starke Sitzungsschlüssel dient der Absicherung der anschließenden Kommunikation.
>
> [...]
>
> Das Protokoll der aktiven Authentisierung sieht regulär vor, eine vom Lesegerät generierte Zufallszahl an den Chip zu übermitteln. Diese Zahl wird dann vom Chip mit dem privaten Schlüssel signiert und an das Lesegerät zurückgeschickt, wodurch ein Nachweis über die Authentizität des Chips erfolgt. Ein modifiziertes Lesegerät könnte hingegen keine „echte" Zufallszahl übermitteln, sondern eine Zahl, die sich z.B. als Hashwert aus Uhrzeit, Ort etc. Ergibt, aber wie eine Zufallszahl aussieht. Mit der dann vom Chip signierten Zahl könnte der Lesegerätbetreiber Dritten gegenüber den Beweis über den Zugriff auf den Pass und damit z.B. den Nachweis des Grenzübertritts führen.
>
> [...]
>
> [Bei der Chip Authentication] erfolgt kein „Unterschreiben" der dem Chip vorgelegten Blöcke." ([Kuegler und Naumann 2007])

Diese als Abstreitbarkeit bezeichnete Funktionalität gilt insbesondere aus Datenschutzperspektive als wichtiger Bestandteil von Extended Access Control.

Die Mitglieder der EU haben sich verpflichtet, Chip Authentication für alle Lesegeräte einzusetzen.

Extended Access Control besteht im Wesentlichen aus 3 Protokollen. Das **Password Authenticated Connection Establishment (PACE)** ist ein Verfahren zur Freischaltung des Chips durch ein Lesegerät. Mit Hilfe einer PIN, die als Passwort fungiert, wird einem Lesegerät der Zugriff auf den Funkkanal zu einem Chip, z.B. dem ePA, gestattet. Aus kryptographischer Perspektive handelt es sich dabei um eine verschlüsselte Diffie-Hellman-Schlüsseleinigung. Die Protokollspezifikation von PACE befindet sich in [BSI-TR-03110], Abschnitt 4.2.1.

Im Rahmen der sog. **Terminal Authentication** prüft der Chip mit Hilfe eines Challenge-Response-Verfahrens die Zugriffsberechtigungen des Lesegeräts (Terminal). Die Protokollspezifikation der Terminal Authentication ist in [BSI-TR-03110], Abschnitt 4.4.1 zu finden.

Die **Chip Authentication** ist ein Verfahren, das genutzt wird, um die Authentizität des Chips zu prüfen. Hierbei werden implizit die Daten, die ein Chip liefert authentisiert. Aus kryptographischer Sicht handelt es sich um eine Diffie-Hellman-Schlüsseleinigung mit statischem Chip-Schlüssel. Nach Abschluss der Chip Authentication ist ein verschlüsselter Ende-zu-Ende-Kanal etabliert. Die Protokollspezifikation der Chip Authentication ist in [BSI-TR-03110], Abschnitt 4.3.1 zu finden.

2.1.3 Data Groups – Speicherplätze für Daten auf dem RF-Chip

Der elektronische Personalausweis enthält personenbezogene Daten in sog. Data Groups. Die Data Groups der sog. eID-Anwendung (siehe auch 2.2.2) sind in Tabelle 1 dargestellt.

Identifikator der Data Group	Inhalt
DG1	Document Type
DG2	Issuing State
DG3	Document Number
DG4	Date of Issue
DG5	Date of Expiry
DG6	Issuing Authority
DG7	**Given Names**
DG8	**Family Names**
DG9	**Maiden Name**
DG10	**Date of Birth**
DG11	**Place of Birth**
DG12	**Nationality**
DG13	**Sex**
DG14	**Eye Color**
DG15	**Height**
DG16	Optional Data
DG17	Normal Place of Residence
DG18	Religious/Artistic Name
DG19	Academic Title
DG20	Residence Permit
DG21	Optional Data

Tabelle 1: Liste der Data Groups mit hervorgehobenen, personenbezogenen Daten für den elektronischen Personalausweis (nach Tabelle E.1 aus [BSI-TR-03110])

Auch der elektronische Personalausweis speichert Daten in Data Groups. Die Liste der Data Groups unterscheidet sich zwischen Reisepass und elektronischem Personalausweis. Für den elektronischen Reisepass ist ein Auszug aus der Liste der Data Groups in der folgenden Tabelle zusammengestellt:

Kontext	Identifikator der Data Group	Inhalt
MRZ	DG1	Alle Daten der machine readable zone (MRZ)
Kodierte, elektronische Identifikationsmerkmale	DG2	Kodiertes, biometrisches Gesichtsbild
	DG3	Biometrischer Fingerabdruck
	DG4	Biometrische Daten der Iris
Visuelle Identifikationsmerkmale	DG5	Gesichtsbild
	DG6	reserviert
	DG7	Angezeigte Unterschrift
Kodierte Sicherheitsmerkmale	DG8	Speicherplatz für Sicherheitsmerkmale
	DG9	Speicherplatz für Sicherheitsmerkmale
	DG10	Speicherplatz für Sicherheitsmerkmale
(kein Kontext)	DG11	Zusätzliche personenbezogene Angaben
	DG12	Zusätzliche Angaben zum Ausweisdokument
	DG13	Weitere, beliebige Details
	DG14	reserviert
	DG15	Speicherplatz für Daten zur aktiven Authentication
	DG16	Im Notfall zu benachrichtigende Personen

Tabelle 2: Liste der Data Groups für den elektronischen Reisepass (in Anlehnung an Figure V-2 aus [ICAO-MRTD-LDS])

Verweise auf bestimmte Data Group IDs müssen also immer kontextabhängig (ePass oder ePA) aufgelöst werden.

2.2 Funktionen des ePA

Die 3 Funktionen des ePA (mitunter synonym als Anwendungen bezeichnet) gliedern sich in die zwei Pflichtfunktionen **ePass** und **eAuthentisierung** sowie eine optionale fortgeschrittene/qualifizierte Signaturfunktion (vgl. [Reisen 2008]). Letztere wird in dieser Arbeit außen vor gelassen, da es sich um eine optionale Komponente handelt und hierzu noch keine Spezifikation der Umsetzung für den ePA vorliegt.

2.2.1 Die ePass-Funktion

Die ePass-Funktion des ePA erlaubt die Anwendung des ePA als elektronischen Reisepass gemäß der EU-Verordnung Nr. 2252/2004. Es

ermöglicht Bürgern des deutschen Staates die einfache Ein- und Ausreise im Schengenraum sowie weiteren ausgewählten Reisezielen im außereuropäischen Raum. Dieser Anwendungsfall – und damit auch der Zugriff auf biometrische Merkmale – bleibt lediglich dem hoheitlichen Bereich vorbehalten. Grundsätzlich sind die Anwendungen strikt getrennt voneinander zu betrachten.
Der deutsche *elektronische Reisepass* enthält derzeit lediglich die ePass-Funktion. Im Falle des *elektronischen Personalausweises* wird aller Voraussicht nach – wie oben dargestellt – die ePass-Funktion eine von maximal 3 Funktionen sein. Auf die ePass-Funktion wird im Rahmen dieser Arbeit nicht weiter eingegangen.

2.2.2 Die Authentisierungsfunktion

Die Authentisierungsfunktion ermöglicht eine sichere Übertragung von Daten, die auf dem Ausweis gespeichert sind an einen Dritten (vgl. [Reisen 2008]). Synonym für den Begriff Authentisierungsfunktion werden die folgenden Bezeichnungen verwendet:

- eAuthentisierung
- (eID) Web-Authentisierung / (eID) Web Authentication
- Online-Authentisierung
- eID Authentication
- eID Application / eID-Anwendung

Spezifische Anwendungen der Authentikationsfunktion sind die Altersverifikation (auch Jugendschutzfunktion) sowie der sektorspezifische Identifikator (siehe Abschnitt 2.3.5).

Die Authentisierungsfunktion soll sowohl für eGovernment als auch für eBusiness verfügbar sein. Andreas Reisen – der für den ePA zuständige Ministerialrat im Bundesministerium des Innern – legt dies wie folgt dar.

> „Der neue Ausweis wird es also Diensteanbietern aus Wirtschaft und Verwaltung gleichermaßen ermöglichen, künftig auch solche Services im Internet aufzusetzen, die aus Nachweis- oder Sicherheitsgründen eine sichere Identitätsfeststellung erfordern und bisher nicht online bereitgestellt werden

konnten.“ ([Reisen 2008], S.164)

Die elektronische Authentisierung ermöglicht damit einerseits völlig neue Anwendungen. Andererseits vereinfacht sie bestehende Angebote und sorgt für ein höheres Maß an Sicherheit und Benutzerfreundlichkeit.
Zu den neu ermöglichten Diensten gehören unter anderem die Online-An- und Ummeldung von Kraftfahrzeugen, die Online-Beantragung von BAFöG, möglicherweise die Abgabe von Steuererklärungen oder Steuervoranmeldungen sowie die Jugendschutzfunktion. Eine hohe Bedeutung kommt dem vereinfachten Identity Management durch den elektronischen Personalausweis zu. Mit Hilfe des ePA können Registrier- und Login-Verfahren einfacher und sicherer gestaltet werden. Heutzutage werden bei vielen Anwendungen eigene Accounts angelegt, die wiederum durch ein individuelles Benutzername/Passwort-Paar geschützt sind. Die Fülle an Authentikations-informationen führt jedoch häufig gerade nicht zu einem sicheren, benutzerfreundlichen Umgang mit diesen Daten. Diese Prozesse können auf die elektronische Authentisierungsfunktion des ePA abgestützt werden. Im Endeffekt besteht somit die Möglichkeit, weit verbreitete Anwendungen wie z.B. Online-Banking und Online-Shopping sicherer und benutzerfreundlicher zu machen.

Aber auch im Offline-Bereich gibt es Vorteile durch die Authentisierungsfunktion des elektronischen Personalausweises. So könnte – wie oben bereits angeführt – beim Hotel-Check-In die Authentisierung elektronisch ohne Preisgabe aller personenbezogenen Daten erfolgen. Typischerweise ist ein Namens- und Adressabgleich beim Check-In gewünscht, demgegenüber müssen Körpergröße und Augenfarbe in der Regel nicht bekannt gegeben werden. An dieser Stelle hilft also die elektronische Authentisierung, lediglich eine Mindestmenge an Informationen preisgeben zu müssen. Dies entspricht dem Grundsatz der Datensparsamkeit und Datenvermeidung des deutschen Datenschutzrechts.
Ferner lässt sich die Jugendschutzfunktion auch im Offline-Bereich umsetzen. Unter der Jugendschutzfunktion versteht man die Prüfung eines Mindest- oder Maximalalters ohne Preisgabe des tatsächlichen Alters der zu überprüfenden Person. Auf diese Weise könnte beispielsweise für Automaten z.B. beim

Zigarettenverkauf oder Video-Verleih die Einhaltung des Jugendschutzes gewährleistet werden.
Insbesondere jedoch bei Online-Diensten stellt die Einhaltung des Jugendschutzes Diensteanbieter vor große Probleme. In Deutschland gelten für derartige Dienste die Anforderungen des Jugendmedienschutzes.

> „In Deutschland ansässige Anbieter müssen zuverlässige Altersverifikationssysteme betreiben, um nicht gegen § 4 Abs. 2 JMStV, § 15 Abs.2 JuSchG und § 184 StGB zu verstoßen. An diese werden in der Rechtsprechung hohe Anforderungen gestellt.“ ([Roßnagel et. al 2008], S. 169)

Derzeit setzen einige Pornografie-Webseiten-Betreiber keine ausreichenden Schutzmechanismen ein. So folgte unter anderem die Arcor AG & Co. KG im September 2007 einem freiwilligen Aufruf zur Sperrung der Pornografie-Webseite youporn.com, da hier kein Altersverifikationssystem im Einsatz war. Darüber hinaus gibt es eine Vielzahl an Webseiten, die keine ausreichende Altersverifikation vornehmen. Hier kam beispielsweise die Personalausweisnummer zum Einsatz.[4] Ein rechtlich anerkanntes Altersverifikationssystem ist beispielsweise das PostIdent-Verfahren oder in Zukunft die elektronische Altersverifikation mit Hilfe des elektronischen Personalausweises.
Im Rahmen der Online-Authentikation erfolgt kein Zugriff auf die biometrischen Merkmale. Sie sind lediglich in der ePass-Anwendung verfügbar. Trotzdem kommt bei der Authentisierungsfunktion Extended Access Control zum Einsatz, da hier zwar keine biometrischen Daten, aber sehr wohl personenbezogene Daten übertragen werden können.
Ein weiterer Unterschied zwischen ePass und ePA ist die Tatsache, dass der ePass in der Regel in einer sicheren Umgebung benutzt wird. Die Authentisierungsfunktion des ePA soll hingegen über das Internet, also einer eher unsicheren Umgebung, verwendet werden können.

4 Die Rechtmäßigkeit der Nutzung von Personalausweis-Seriennummern zur Verifikation ist vor dem Hintergrund der Gefahr der Profilbildung durch Verwendung eines einheitlichen Personenkennzeichens fraglich.

2.3 Datenschutzaspekte des elektronischen Personalausweises

Dieser Abschnitt diskutiert Datenschutzaspekte des elektronischen Personalausweises. Die Einführung des elektronischen Personalausweises stellt eine grundlegende Änderung des vermutlich bedeutendsten Ausweisdokuments in Deutschland dar. Aufgrund des strengen Datenschutzrechts in Deutschland, müssen die neuen Funktionen des ePA auf ihre Konformität mit dem Datenschutzrecht geprüft werden. Hierzu wurde im Herbst 2007 ein Gutachten durch Prof. Dr. Alexander Roßnagel, Gerrit Hornung und Christoph Schnabel angefertigt (vgl. [Roßnagel et. al 2008]). Die wichtigsten Kernaussagen sind im Folgenden zusammengefasst. Dies ersetzt naturgemäß nicht die vollständige Ausführung des Gutachtens und weist auch nicht in annähernd äquivalenter Form eine juristische Begutachtung auf, sondern hilft lediglich die grobe Konformität mit dem Datenschutzrecht zu beurteilen. Da die Authentisierungsfunktion im Mittelpunkt dieser Arbeit steht, beschränkt sich dieser Abschnitt auf ihre Beurteilung, die ePass-Funktion sowie die optionale Signaturfunktion werden hier nicht näher betrachtet.

2.3.1 Freiwilligkeitsprinzip und bewusste, zielgerichtete Handlung

Zunächst ist das Freiwilligkeitsprinzip der Authentisierungsfunktion hervorzuheben. Der Ausweisinhaber kann selbst freiwillig entscheiden, ob er die Authentisierung nutzen möchte. Sie wird zwar standardmäßig bereitgestellt, kann jedoch bei Abholung des Ausweises durch einen entsprechenden Antrag deaktiviert werden. Die o.g. Juristen beurteilen dies positiv.

> „Die Freiwilligkeit der Anwendung ist im Sinne der informationellen Selbstbestimmung vorbildlich.“ ([Roßnagel et. al 2008])

Ein ähnlicher Grundsatz gilt für die Durchführung der Authentisierung. Hier muss dem Benutzer deutlich werden, dass er

> „[...] in einer bewussten und zielgerichteten Handlung über den Einsatz des Identitätsnachweises, also der Übermittlung personenbezogener Daten an Dritte, entscheiden kann. Dies wurde bei der Konzeption der Authentisierungsfunktion berücksichtigt, indem für die Übermittlung von personenbezogenen Daten an

> Dritte der Personalausweis immer in physischer Form vorliegen und eine Geheimzahl (PIN) eingegeben werden muss. Es ist also immer eine Handlung des Ausweisinhabers bzw. der Ausweisinhaberin erforderlich.“ ([Reisen 2008])

Als ein entscheidendes Kriterium bleibt folglich die bewusste, zielgerichtete Handlung, die der Ausweisinhaber durchführen muss.

2.3.2 Kontrolle und Ahndung

Bei der Authentisierungsfunktion muss die auslesende Instanz, das sog. Terminal, die Berechtigung zum Auslesen von personenbezogenen Daten besitzen. Dies wird in Form von Zertifikaten sichergestellt. Das Zugriffszertifikat, das sog. Terminal Certificate, wird von einer Datenschutzaufsichtsbehörde ausgegeben.

> „Für die Kontrolle personenbezogener Datenverarbeitungen und die Ahndung von Verstößen gegen die datenschutzrechtlichen Vorgaben sind bekanntlich der Bundesbeauftragte für Datenschutz und Informationsfreiheit, die 16 Landesdaten-schutzbeauftragten und die 16 Datenschutzaufsichtsbehörden im nicht-öffentlichen Dienst zuständig.“ ([Reisen 2008])

Details über die Art und Weise der Prüfung folgen in Abschnitt 2.3.4 Plausibilitätsprüfung und Zugriffserlaubnis.

2.3.3 Transparenz

Darüber hinaus muss vor jeder Transaktion im Rahmen der Authentisierung eine definierte Menge an Informationen dargestellt werden. Es ist technisch sicherzustellen, dass diese Informationen zusammengehörig und gut sichtbar dargestellt werden. Teil dieser Menge sind mindestens die folgenden erforderlichen Informationen:

- die genaue Bezeichnung der auslesenden Instanz,
- die auszulesenden Datenfelder des ePA,
- der Verwendungszweck der Daten sowie
- die zuständige Datenschutzaufsichtsbehörde.

Für den Fall, dass ein Benutzer einen Datenmissbrauch vermutet, kann auf diese Weise direkt die zuständige Datenschutzaufsichtsbehörde eruiert und

kontaktiert werden.

2.3.4 Plausibilitätsprüfung und Zugriffserlaubnis

Bevor die Zertifikate zum Auslesen von Daten im Rahmen der Authentisierung ausgegeben werden, muss geprüft werden, ob die Erhebung und Verwendung der Daten rechtskonform ist. Diese Aufgabe übernehmen – wie oben geschildert – die Datenschutzaufsichtsbehörden. Derzeit sieht das Konzept vor, dass eine Plausibilitätsprüfung der datenschutzrechtlichen Erforderlichkeit durchgeführt werden soll. Nach den Gutachtern verbirgt sich in der Praxis dahinter in vielen Fällen hauptsächlich die Entscheidung, ob eine

> „[...] eineindeutige Identifizierung des Ausweisinhabers erforderlich ist oder nur bestimmte Attribute wie Volljährigkeit oder Wohnort benötigt werden, die auch anonym übermittelt werden können.
>
> [...]
>
> Die Prüfung der Erforderlichkeit des Zugriffs auf alle Felder der Authentisierungsfunktion muss sich auch an dem Gebot des § 3a BDSG zur Datenvermeidung und Datensparsamkeit und – für Telemedien – dem Gebot des § 13 Abs. 6 TMG, die anonyme und pseudonyme Inanspruchnahme und Bezahlung von Telemedien zu ermöglichen, ausrichten.“ ([Roßnagel et. al 2008])

Die Plausibilitätsprüfung gewährleistet also die Einhaltung des Grundsatzes zur Datenvermeidung und Datensparsamkeit, bevor ein Zertifikat ausgegeben wird. Ferner sollte eine anonyme oder pseudonyme Nutzung eines Dienstes ermöglicht werden, wenn der Dienst mit derselben Qualität erbracht werden kann.

2.3.5 Vermeidung von Personenkennzeichen

Ein möglicher Anwendungsfall der Authentisierungsfunktion des ePA ist die Verwendung als Anmelde-Berechtigungsnachweis anstelle eines Benutzername/Passwort-Paars. Die hierfür notwendige Eindeutigkeit des

individuellen ePA scheint zunächst im Widerspruch zum Datenschutz zu stehen. Es ist gesetzlich verboten, die Seriennummer des Ausweises oder jegliche vergleichbare global eindeutige Nummer zu erheben (vgl. [Roßnagel et. al 2008]). Aus diesem Grund erfordern einige Online-Dienste die Preisgabe der eineindeutigen Kombination von personenbezogenen Attributen, d.h. der volle Name, Geburtsort und Geburtsdatum. Der ePA kann dies umgehen, indem für einen Diensteanbieter ein sog. **sektorspezifischer Identifikator** erstellt wird. Dieser sektorspezifische Identifikator wird aus einer Kennung des Diensteanbieters sowie einer Kennung des Personalausweises von dem ePA unter Zuhilfenahme kryptographischer Funktionalität generiert.[5] Die Eindeutigkeit ist jedoch absichtlich nur für diesen einen Diensteanbieter gewährleistet. Darüber hinaus ist es nicht möglich, dass 2 Diensteanbieter die sektorspezifischen Identifikatoren sinnvoll untereinander abgleichen. Ferner verhindert das vom elektronischen Personalausweis verwendete Verfahren, dass anhand des sektorspezifischen Identifikators Rückschlüsse auf die Ausweisinhaberin bzw. den Ausweisinhaber gezogen werden können bzw. deren oder dessen Identität ermittelt werden kann.

5 Auf eine detaillierte Beschreibung des Protokolls wird hier aus Platzgründen verzichtet. Die Spezifikation findet sich in [BSI-TR-03110], Abschnitt 4.5

3 Web-Authentisierung – Einsatzszenario der Authentisierungsfunktion

Diese Arbeit umfasst im Folgenden zwei Hauptaspekte. Zum einen wird in Kapitel 3 die Nutzung des elektronischen Personalausweises zur Authentisierung im Internet dargestellt. Hierbei werden Anforderungen definiert und im darauf folgenden Kapitel 4 die notwendigen Komponenten beschrieben. Zum anderen beinhaltet diese Abschlussarbeit eine Beispiel-Implementierung der Nutzung des elektronischen Personalausweises zur Authentisierung von Attributen gegenüber einem Web-Portal. Unter einer Attributsauthentikation versteht man die sichere Abfrage von Merkmalen. Ein Merkmal ist beispielsweise das Alter einer Person oder personenbezogene Daten wie Vor- und Nachname. Details zu den erforderlichen Abläufen sowie eine Beschreibung der Implementierung folgen in den Kapiteln 5 und 6. Beispielhaft wird in diesem Kapitel ein Szenario beschrieben, in dem sich ein Benutzer mit Hilfe des elektronischen Personalausweises (ePA) gegenüber einer Webseite authentisiert und auf sicherem Weg personenbezogene Daten übermittelt.

Aus Sicht eines Webseiten-Betreibers bedeutet diese Web-Authentisierung eine verlässliche Möglichkeit der Verifikation, mit welcher Person tatsächlich interagiert wird. Im Rahmen der Web-Authentisierung werden beispielsweise Vor- und Nachname eines Kunden verifiziert. Es ist zu beachten, dass es sich hierbei immer um Daten handelt, die vom Kunden entweder bereits freiwillig angegeben wurden oder die im Rahmen eines Geschäftsvorgangs unbedingt angegeben werden müssen. Im Falle einer Online-Bestellung werden beispielsweise Vor- und Nachname bereits heute durch den Webshop-Betreiber erfasst und in der Regel in Kombination mit einer Adresse für die Lieferung des Produkts benötigt und verwendet.[6]

Alle in dieser Arbeit aufgezeigten Vorgänge und Komponenten orientieren sich

6 Die Menge an Informationen, die im Rahmen der Authentisierung zugänglich gemacht werden, hängt von den Berechtigungen des Terminal-Zertifikats (die wiederum vom Ausgang der vorhergehenden Plausibilitätsprüfung abhängt) ab. Siehe hierzu auch Abschnitt 4.7.4 (Terminal Certificate (TC)) ab Seite 60.

in erster Linie an der Technischen Richtlinie BSI TR 03110 des Bundesamtes für Sicherheit in der Informationstechnik mit dem Titel „Advanced Security Mechanisms for Machine Readable Travel Documents – Extended Access Control (EAC)“ in der Version 2.0 (siehe [BSI-TR-03110]).

3.1 Authentisierung mit einem Web-Portal

Die folgende Abbildung zeigt die Komponenten, die in dem folgenden Beispiel-Szenario verwendet werden.

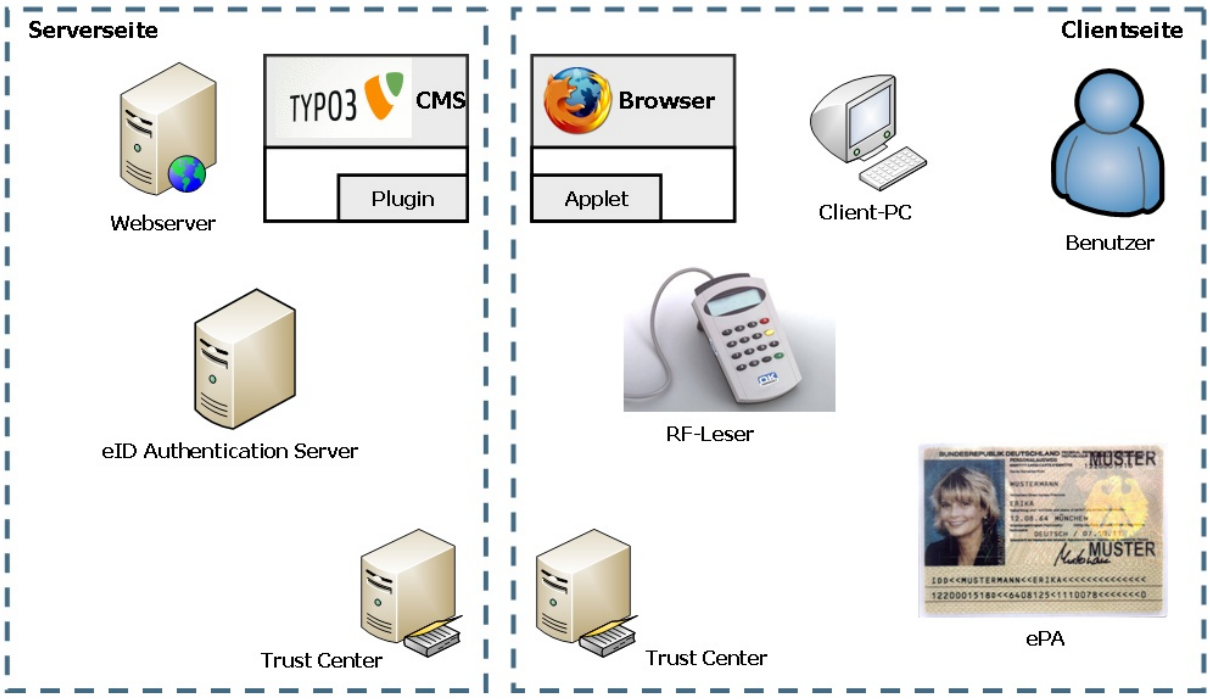


Abbildung 9: Komponenten und Aufbau des Beispiel-Szenarios

Abbildung 9 zeigt die beteiligten Komponenten für das Szenario, bei dem sich ein Benutzer (in der Abbildung rechts) gegenüber einer Webseite oder einem Web-Portal (links) authentifizieren und auf sichere Art und Weise personenbezogene Daten übermitteln möchte. Darüber hinaus findet eine Authentifizierung der Webseite gegenüber dem Benutzer statt, damit dieser nachvollziehen kann, welche Instanz seine Daten abfragt. Es handelt sich also um eine gegenseitige Authentisierung.

Die Abbildung ist in zwei horizontal angeordnete, durch gestrichelte Rechtecke gekennzeichnete Bereiche, sog. Komponentengruppen, eingeteilt. Auf der linken Seite wird der Aufbau einer Webseite bzw. eines Web-Portals dargestellt. Die rechte Seite repräsentiert die Infrastruktur des Benutzers.
Im Fall des Benutzers handelt es sich in diesem Beispiel-Szenario um einen Bürger mit einem deutschen elektronischen Personalausweis. Der Webseiten-Betreiber ist ein (nicht notwendigerweise kommerzieller) Online-Anbieter mit einem Web-Angebot wie beispielsweise einem Online-Versandhaus. Die Tatsache, ob es sich um ein kommerzielles Angebot handelt, spielt jedoch für die technische Realisierung keine Rolle.

Auf der Seite des Benutzers (in Abbildung 9 rechts) befinden sich die folgenden Komponenten:

- der elektronische Personalausweis (ePA) des Benutzers
- ein Client-PC mit einem Webbrowser (z.B. Mozilla Firefox)
- ein RF-Lesegerät, das an den Client-PC angeschlossen ist
- eine Software-Komponente, die mit dem RF-Lesegerät kommuniziert (z.B. ein Applet oder ein Browser-Plugin)

Auf der Seite des Webseiten-Betreibers (in Abbildung 9 links) befinden sich die folgenden Komponenten:

- der Webserver
- ein Content-Management-System (CMS) (z.B. Typo3)
- ein Proxying-Plugin im CMS
- der eID Authentication Server mit einem (Hoch-)Sicherheitsmodul

Beide Komponentengruppen referenzieren Trust Center. Sie garantieren die Echtheit sowohl des Ausweisdokumentes als auch des Webseiten-Betreibers. Diese Trust Center sind in eine komplexe Public Key Infrastructure eingebunden.[7]

Die Komponenten und ihre Funktionen werden in Kapitel 4 (Komponentenbeschreibung) ab Seite 36 im Detail beschrieben.

7 Für eine detaillierte Beschreibung der Komponente PKI siehe Abschnitt 4.2 (Public Key Infrastructure und Key Management)

3.2 Ablauf der Web-Authentisierung

In diesem Szenario soll in erster Linie gezeigt werden, wie der Webseiten-Betreiber sicher mit dem elektronischen Personalausweis eines Benutzers kommunizieren kann. Im Rahmen der sicheren Kommunikation sollen bestimmte, vorher definierte Datenfelder des ePA durch den Webseiten-Betreiber ausgelesen werden können. Der Ablauf des Zugriffs auf den ePA gliedert sich dabei in die folgenden vier Phasen:

1. INIT: Der Benutzer signalisiert der Webseite, dass er die Web-Authentisierung mit dem ePA wünscht.
2. SETUP: Zwischen dem eID Authentication Server (EAS) auf der Serverseite und dem elektronischen Personalausweis (ePA) auf der Clientseite wird mit Hilfe der Extended Access Control Mechanismen der BSI TR 03110 ein sicherer Kanal aufgebaut.
3. READ: Nach erfolgreichem Aufbau des sicheren Kanals werden die Datenfelder des ePA vom Webseiten-Betreiber ausgelesen.
4. CLOSE: Schließlich wird der sichere Kanal abgebaut.

Diese Phasen werden in den Abschnitten 5.2 bis 5.5 detailliert erläutert.

Die ausgelesenen Daten wie zum Beispiel der volle Name und die Adresse können dann auf der Serverseite beispielsweise dazu genutzt werden, um eine Bestellung in einem Webshop abzuschließen oder ein Login auf einer Webseite durchzuführen.

Der o.g. Ablauf der Phasen wurde im Rahmen einer Beispiel-Implementierung umgesetzt. Details zur Implementierung finden sich in den folgenden Kapiteln. Die Beispiel-Implementierung zeigt, dass nach einem erfolgreichen Abschluss der o.g. vier Schritte der Webseiten-Betreiber einen sicheren Zugriff auf die Attribute hat. Der Ablauf könnte beispielsweise in den folgenden Workflow eingebunden werden:

1. Der Benutzer besucht einen Webshop und stellt einen Warenkorb zusammen.
2. Der Webshop-Betreiber bietet dem Benutzer an, die eID Web-Authentisierung durchzuführen.
3. Der Benutzer initiiert die Web-Authentisierung, übermittelt Vor-,

Nachname sowie Adresse und schließt die Bestellung ab. Verläuft die Web-Authentisierung erfolgreich, so hat der Webshop-Inhaber die Gewissheit über die Integrität und Authentizität der entsprechenden Attribute.

Im Detail soll der Benutzer im Rahmen der Beispiel-Implementierung die folgenden Schritte durchlaufen:

1. Der Benutzer soll zunächst eine Webseite mit einem Formular mit den folgenden Feldern aufrufen:
   - Nachname
   - Vorname
   - Geburtsort
   - Geschlecht
2. Der Benutzer übermittelt die Daten aus seinem elektronischen Personalausweis mit Hilfe der Authentikationsfunktion unter Verwendung von Extended Access Control.
3. Die Daten werden zur Kontrolle in dem Formular angezeigt.

Dieser Ablauf wird anhand von Screenshots der Beispielimplementierung in Abschnitt 6.10 ab Seite 102 dokumentiert.

Das Beispiel-Szenario ist bewusst sehr flexibel gehalten und nicht auf eine konkreten Workflow beschränkt. Das Szenario passt damit sowohl in einen typischen eGovernment-Workflow, wie beispielsweise einer Online-Anmeldung eines KFZ, als auch in einen eBusiness-Ablauf, wie z.B. im Rahmen eines Bestellvorgangs in einem Webshop. Der universelle Charakter soll deutlich im Vordergrund stehen.

4 Komponentenbeschreibung

Dieses Kapitel beschreibt die benötigten Komponenten für die Beispiel-Implementierung der Authentikationsfunktion. Sie werden nicht nur modellhaft beschrieben, sondern bei Bedarf mit Beispielen veranschaulicht.

> **Bemerkung**
> Es gibt zu manchen Komponenten Aspekte, die aufgrund des Beispielcharakters der begleitenden Implementierung nicht oder nur teilweise realisiert werden. Wenn Komponenten vereinfacht implementiert werden, so ist dies an entsprechender Stelle in diesem Dokument vermerkt.

Die grundlegende Spezifikation für alle Komponenten bildet die technische Richtlinie BSI TR 03110 „Advanced Security Mechanisms for Machine Readable Travel Documents – Extended Access Control (EAC)“ in der Version 2.0 – Public Beta 2 [BSI-TR-03110].
Im Folgenden wird davon ausgegangen, dass für die kryptographischen Verfahren lediglich Kryptographie der Elliptischen Kurven (ECC) zum Einsatz kommt, selbst wenn die Spezifikationen grundsätzlich auch andere Verfahren zuließen.[8]
Neben der Beschreibung der Bestandteile der Komponenten werden Handlungsanweisungen in Form von Ablaufbeschreibungen für eine Implementierung gegeben. Die Handlungsanweisungen orientieren sich dabei im Wesentlichen an den folgenden weiteren technischen Richtlinien des BSI:

- BSI TR 03111 „Elliptic Curve Cryptography based on ISO 15946“, Version 1.00 (siehe [BSI-TR-03111])
- BSI TR 20102 „Kryptographische Verfahren: Empfehlungen und Schlüssellängen“, Version 0.7 (siehe [BSI-TR-20102])
- Brainpool Group „ECC Brainpool Standard Curves and Curve Generation“,

8 Die BSI TR 03110 ist nicht auf Deutschland beschränkt und beinhaltet daher auch Verfahren, die nicht mit ECC realisiert werden. Für die Anwendung der Technischen Richtlinie in Deutschland gilt allerdings, dass lediglich ECC verwendet werden sollte.

Version 1.0 (siehe [BrainpoolECC])

- BSI TR 03116 „Technische Richtlinie für die eCard-Projekte der Bundesregierung“, Version 1.0 (siehe [BSI-TR-03116]) sowie
- ICAO Technical Report „PKI for Machine Readable Travel Documents offering ICC Read-Only Access“, Version 1.1 (siehe [ICAO-PKI-MRTD-ICC]).

Bei Bedarf wird an der entsprechenden Stelle auf weitere Dokumente verwiesen.

4.1 Architektur

Die folgende Abbildung zeigt in Anlehnung an Abbildung 9 die Architektur des Gesamtaufbaus. Hieraus sind insbesondere die Schnittstellen der Komponenten untereinander (dunkelrote, durchgezogene Linie) ersichtlich.

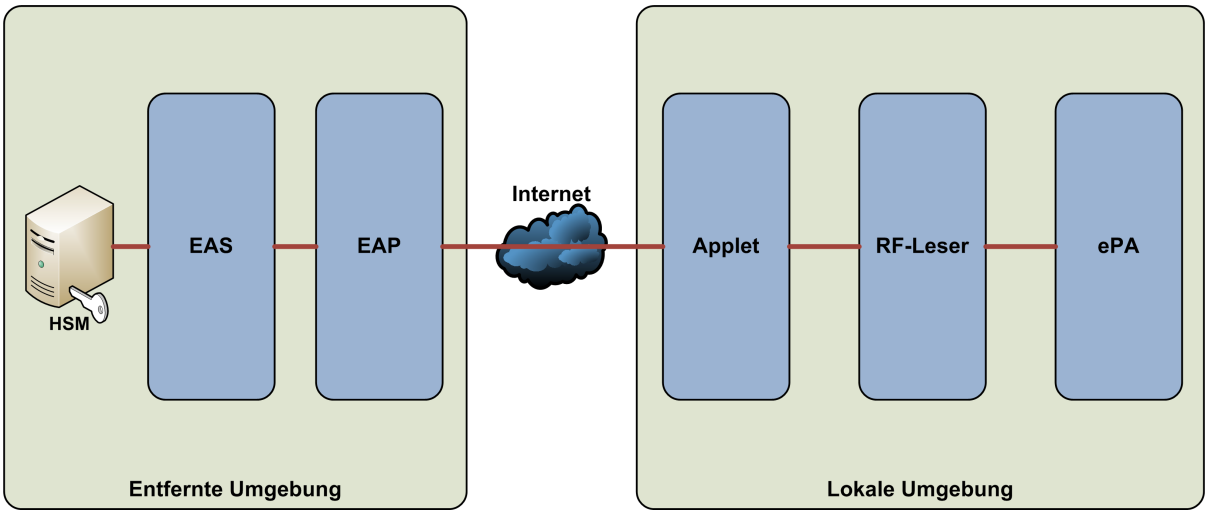


Abbildung 10: Architektur des Gesamtaufbaus

Abbildung 10 dient auch als Referenzgrafik für die Beschreibung der Schnittstellen innerhalb der Komponentenbeschreibung. Die Komponenten Applet, RF-Leser und ePA befinden sich in der lokalen Umgebung des Benutzers. Üblicherweise werden alle Komponenten der lokalen Umgebung auf oder an dem Computer des Anwenders zu finden sein.

Die Komponenten eID Authentication Server und eID Authentication Proxy befinden sich in der entfernten Umgebung. Diese Komponenten müssen nicht zwangsläufig auf ein und demselben Computer zu finden sein. Es ist sogar empfehlenswert, sie aus Sicherheitsgründen auf unterschiedliche Computern zu verteilen.

4.2 Public Key Infrastructure und Key Management

Im Rahmen der eID-Anwendung werden diverse Schlüsselpaare benötigt. Auf der einen Seite benötigt mindestens die zugreifende Instanz, das sog. Terminal, ein Schlüsselpaar. Auf der anderen Seite muss die auszulesende Instanz, der elektronische Personalausweis, ein Schlüsselpaar besitzen.
Die öffentlichen Schlüssel müssen entweder in Form eines Zertifikats oder in einer anderen von einer vertrauenswürdigen Instanz signierten Form vorgehalten werden. Die geheimen Schlüssel sollten in sicherer Form und für Dritte unzugänglich, beispielsweise in einem (Hoch-)Sicherheitsmodul, gespeichert werden.

Die Komponente PKI stellt in einer Beispiel-Implementierung die Funktionalität zur Verfügung, die für das Schlüsselmanagement sowie die Public Key Infrastructure benötigt werden. Hierbei handelt es sich aufgrund des Beispielcharakters der Implementierung nicht um eine vollwertige PKI, sondern es werden lediglich die notwendigen Funktionen realisiert.

4.2.1 Public Key Infrastructure

Die Public Key Infrastructure (PKI) der eID-Anwendung orientiert sich an der PKI der ePass-Anwendung nach [ICAO-PKI-MRTD-ICC]. Man unterscheidet grundsätzlich zwei Teile der PKI.

4.2.1.1 Integrität von Ausweisdokumenten

Ein Teil der PKI kümmert sich um die Ausstellung der Ausweisdokumente und garantiert ihre Echtheit. Dies ist der sog. Signing Part. Im Rahmen der Herstellung und Personalisierung eines elektronischen Ausweisdokuments wird beispielsweise das statische Schlüsselpaar des ePA von einer darüber

liegenden Instanz signiert. Die dahinter stehende PKI (siehe Abbildung 11) besitzt auf nationaler Ebene eine Wurzelinstanz, die sog. **Country Signing Certificate Authority (CSCA)**.

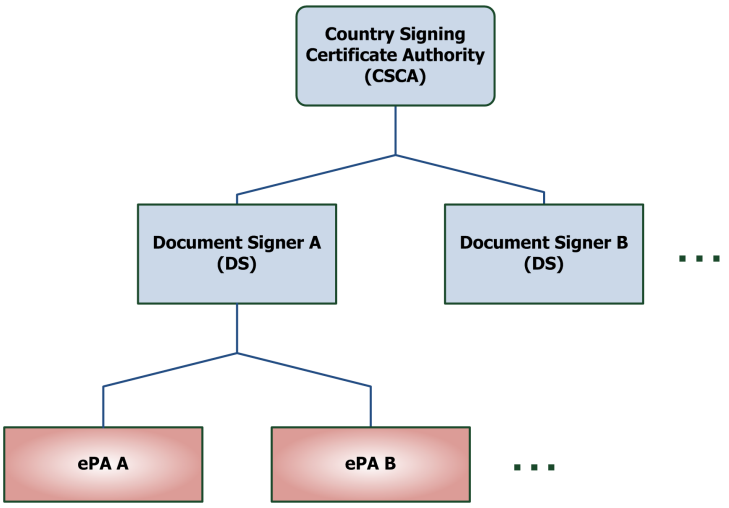


Abbildung 11: Public Key Infrastructure des Signing Part

Unterhalb der CSCA existieren sog. **Document Signer (DS)**. Document Signer haben die Aufgabe, einen elektronischen Personalausweis herzustellen und das statische Schlüsselpaar des ePA zu signieren. Diese Signatur kann dann von auslesenden Instanzen benutzt werden, um die Echtheit des ePA, d.h. die Herausgabe des ePA durch eine Instanz unterhalb der CSCA, zu verifizieren.
Es ist Aufgabe der CSCA, die Zertifikate der Document Signer zu signieren und herauszugeben. Ein vertrauenswürdiges Zertifikat eines Document Signer muss durch die CSCA signiert sein. Die CSCA garantiert somit die Echtheit eines Document Signer.

Für die Beispiel-Implementierung im Rahmen der Web-Authentisierung bedeutet dies, dass der elektronische Personalausweis ein signiertes, statisches Schlüsselpaar besitzt, wobei sich die Vertrauenskette des öffentlichen Schlüssels, mit Hilfe dessen die Signatur überprüft werden kann, lückenlos bis zur Country Signing CA schließen lässt. Üblicherweise führt diese Vertrauenskette der übergeordneten Instanzen von der Country Signing CA zu einem Document Signer und enthält damit lediglich zwei Elemente.

4.2.1.2 Signatur von Terminal Certificates

Ein anderer Teil der PKI, der sog. Verifying Part, bezieht sich auf die Verifikation der Terminals, also der zugreifenden Instanzen (siehe Abbildung 12).

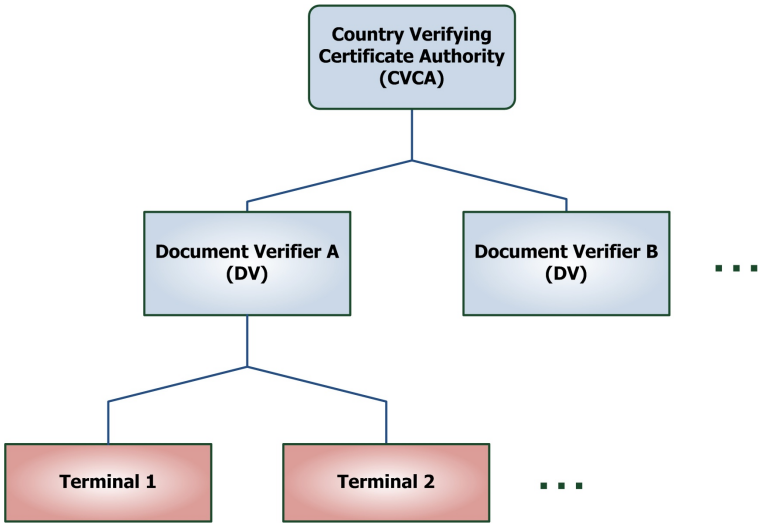


Abbildung 12: Public Key Infrastructure des Verifying Part

Dies bedeutet, dass es auf nationaler Ebene eine weitere Wurzelinstanz, die sog. **Country Verifying Certificate Authority (CVCA)** gibt. Sie unterscheidet sich von der Country Signing CA dadurch, dass sie ausschließlich für die Terminals und nicht für die Ausweisdokumente zuständig ist. CSCA und CVCA haben also unterschiedliche Aufgabenbereiche.

Die folgende Abbildung zeigt ein Beispiel für ein Tag-Length-Value-kodiertes CVCA-Zertifikat für die ePass-Anwendung z.B. auf einem elektronischen Reisepass. Üblicherweise handelt es sich bei den Zertifikaten um sogenannte Card Verifiable Certificates (CV Certificates, dt. karten-verifizierbare Zertifikate). CV Certificates bieten den Vorteil, dass sie selbst in Umgebungen mit geringen Ressourcen, wie z.B. einer SmartCard oder dem Chip eines Ausweises, verifiziert werden können.
Das Zertifikat aus Abbildung 13 bzw. Tabelle 3 ändert sich für die

Authentisierungsfunktion des elektronischen Personalausweises lediglich an den in Tabelle 3 farblich gekennzeichneten Stellen.

Figure F.5.: CVCA Certificate (ECDSA)

```
0000 : 7F218201 897F4E82 01495F29 01004204 44454356 43414550 41535330 30317F49
0020 : 81FD06DA 04007F00 07020202 02028111 D7C134AA 26436686 2A183025 75D1D787
0040 : B09F0757 97DA89F5 7EC8C0FF 821C68A5 E62CA9CE 6C1C2998 03A6C153 0B514E18
0060 : 2AD8B004 2A59CAD2 9F438311 2580F63C CFE44138 870713B1 A92369E3 3E2135D2
0080 : 66DBB372 386C400B 84390400 9029AD2C 7E5CF434 0823B2A8 7DC68C9E 4CE3174C
00A0 : 1E6EFDEE 12C07D58 AA56F772 C0726F24 C6B89E4E CDAC2435 4B9E99CA A3F6D376
00C0 : 1402CD85 1CD7C134 AA264366 862A1830 2575D0FB 98D116BC 4B6DDEBC A3A5A793
00E0 : 9F866390 4393EE8E 06DB6C7F 528F8B426 0B49AA93 309824D9 2CDB1807 E5437EE2
0100 : E26E29B7 3A711153 0FA86B35 0037CB94 15E15370 43944637 97139E14 87010105F
0120 : 200E4445 43564341 45504153 53303031 7F4C0E06 09040070 00070301 02015301
0140 : C15F2506 00070004 00015F24 06000900 040001 5F 37384CCF 25C59F36 12EEE188
0160 : 75F6C5F2 E2D21F03 95683B53 2A26E4C1 89B71EFE 659C3F26 E0EB9AEA E9986310
0180 : 7F9B0DAD A16414FF A204516A EE2B
```

Tag Length Value

Abbildung 13: Farbig markiertes, Tag-Length-Value-kodiertes CVCA-Zertifikat für die ePass-Anwendung nach Figure F.5 aus [BSI-TR-03110]

Das Beispiel-Zertifikat gehört zu einer CVCA und erlaubt den darunter liegenden Instanzen prinzipiell den Zugriff auf den biometrischen Fingerabdruck über Extended Access Control. Die Struktur des Zertifikats wird in Tabelle 3 dargestellt.

Bemerkung

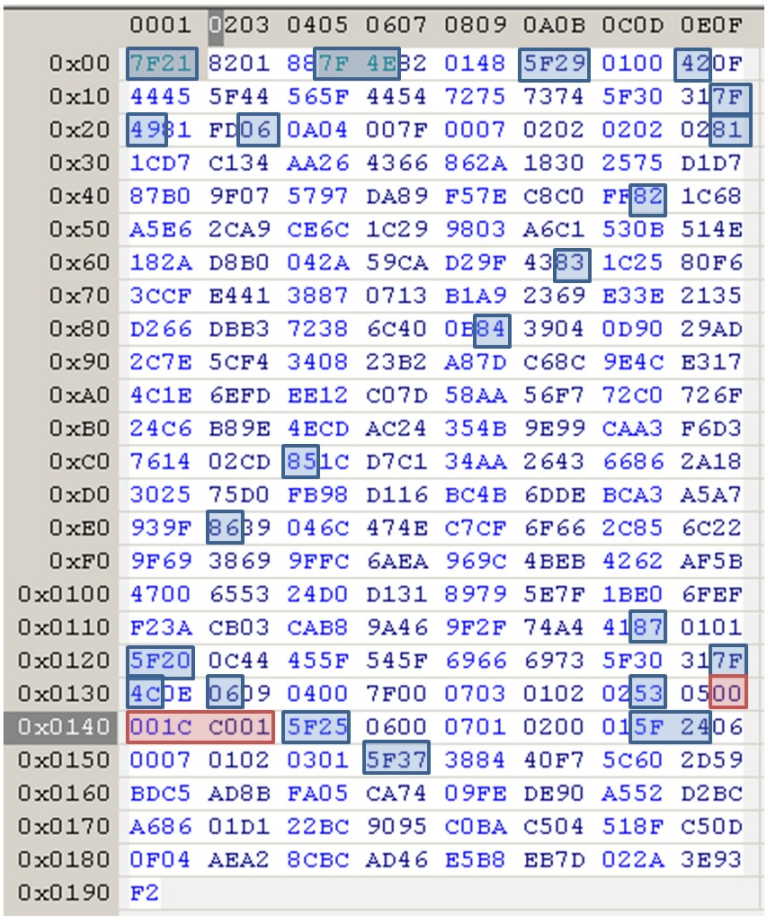
Die in Abschnitt 2.3.3 genannten Merkmale „zuständige Datenschutzaufsichtsbehörde“ sowie „Verwendungszweck der erhobenen Daten“ sind noch nicht implementierbar, da die entsprechende Spezifikation keine Felder für diese Daten vorsieht.

Tag	Bedeutung
0x7F21	CV Certificate
0x7F4E	CV Certificate Body
0x5F29	Certificate Profile Identifier (gibt die Version des Zertifikats bzw. das Format im Falle eines Certificate Requests an)
0x42	Certificate Authority Reference (identifiziert den öffentlichen Schlüssel der ausgebenden CA)
0x7F49	Public Key (hier ein öffentlicher ECDSA-Schlüssel)
0x06	Terminal Authentication with ECDSA-SHA-224
0x81	Prime modulus p
0x82	Coefficient a
0x83	Coefficient b
0x84	Base point G
0x85	Order of the base point r
0x86	Public point Y
0x87	Cofactor f
0x5F20	Certificate Holder Reference (assoziiert einen Bezeichner zu dem öffentlichen Schlüssel dieses Zertifikats)
0x7F4C	Certificate Holder Authorization Template (beinhaltet die Zugriffsberechtigungen des Inhabers dieses Zertifikats auf die Data Groups. Rollen können z.B. CVCA, DV oder Terminal sein.)
0x06	Extended Access Control for ePassports
0x53	Discretionary Data (hier: die Instanz vom Typ CVCA erlaubt den Zugriff auf Data Group 3, Data Group 3 ist im Kontext des ePasses der biometrische Fingerabdruck (siehe Tabelle 2))
0x5F25	Certificate Effective Date (Beginn des Gültigkeitszeitraums dieses Zertifikats)
0x5F24	Certificate Expiration Date (Ende des Gültigkeitszeitraums dieses Zertifikats)
0x5F37	Digital Signature (hier: ECDSA in Plain Format)

Tabelle 3: Dekodiertes CVCA-Zertifikat aus Abbildung 13 für die ePass-Anwendung

An den farblich gekennzeichneten Stellen in Tabelle 3 werden im Falle der Authentisierungsfunktion die Zugriffsberechtigungen auf geringfügig andere Art und Weise kodiert. Für die ePass-Anwendung werden sie lediglich in einem Byte kodiert, für die Authentisierungsfunktion hingegen werden sie in 5 Bytes kodiert. Der Unterschied liegt darin, dass im Fall der ePass-Anwendung lediglich Rollenzugehörigkeit angegeben werden kann, während die Authentisierungsfunktion neben den Rollen auch für jede Data Group einzeln Lese- und Schreibrechte verwalten kann. In Abbildung 13 ist dies daran zu erkennen, dass das Tag **`0x06`** innerhalb von **`0x074C`** den Wert **`0.4.0.127.0.7.3.1.2.1`** trägt, der dem ASN.1 Object Identifier für die ePass-Anwendung entspricht (vgl. Tabelle 6, Seite 61). Für die Authentisierungsfunktion würde hier der Wert **`0.4.0.127.0.7.3.1.2.2`** stehen müssen (vgl. Abbildung 14). Im anschließenden Tag **`0x53`** würden die

Berechtigungen gemäß der Tabelle C.4 aus [BSI-TR-03110] kodiert (siehe auch Abbildung 20). Das Beispielzertifikat für die ePass-Anwendung aus Abbildung 13 hingegen verwendet hierfür Tabelle C.3 aus [BSI-TR-03110]. Die folgende Abbildung zeigt ein kodiertes CVCA-Zertifikat für den ePA:



	0001	0203	0405	0607	0809	0A0B	0C0D	0E0F
0x00	7F21	8201	887F	4E82	0148	5F29	0100	420F
0x10	4445	5F44	565F	4454	7275	7374	5F30	317F
0x20	4981	FD06	0A04	007F	0007	0202	0202	0281
0x30	1CD7	C134	AA26	4366	862A	1830	2575	D1D7
0x40	87B0	9F07	5797	DA89	F57E	C8C0	FF82	1C68
0x50	A5E6	2CA9	CE6C	1C29	9803	A6C1	530B	514E
0x60	182A	D8B0	042A	59CA	D29F	4383	1C25	80F6
0x70	3CCF	E441	3887	0713	B1A9	2369	E33E	2135
0x80	D266	DBB3	7238	6C40	0B84	3904	0D90	29AD
0x90	2C7E	5CF4	3408	23B2	A87D	C68C	9E4C	E317
0xA0	4C1E	6EFD	EE12	C07D	58AA	56F7	72C0	726F
0xB0	24C6	B89E	4ECD	AC24	354B	9E99	CAA3	F6D3
0xC0	7614	02CD	851C	D7C1	34AA	2643	6686	2A18
0xD0	3025	75D0	FB98	D116	BC4B	6DDE	BCA3	A5A7
0xE0	939F	8639	046C	474E	C7CF	6F66	2C85	6C22
0xF0	9F69	3869	9FFC	6AEA	969C	4BEB	4262	AF5B
0x0100	4700	6553	24D0	D131	8979	5E7F	1BE0	6FEF
0x0110	F23A	CB03	CAB8	9A46	9F2F	74A4	4187	0101
0x0120	5F20	0C44	455F	545F	6966	6973	5F30	317F
0x0130	4C0E	0609	0400	7F00	0703	0102	0253	0500
0x0140	001C	C001	5F25	0600	0701	0200	015F	2406
0x0150	0007	0102	0301	5F37	3884	40F7	5C60	2D59
0x0160	BDC5	AD8B	FA05	CA74	09FE	DE90	A552	D2BC
0x0170	A686	01D1	22BC	9095	C0BA	C504	518F	C50D
0x0180	0F04	AEA2	8CBC	AD46	E5B8	EB7D	022A	3E93
0x0190	F2							

Abbildung 14: Hexdump eines CVCA-Zertifikats für die eID-Anwendung im TLV-Format. In blau hervorgehoben sind die Tags, in rot hervorgehoben sind die kodierten Zugriffsberechtigungen.

Unter der CVCA existieren sog. **Document Verifier (DV)**. Document Verifier haben die Aufgabe, so genannte **Berechtigten-Zertifikate** oder **Terminal Certificates (TC)** auszustellen.
Jeglicher Zugriff auf den elektronischen Personalausweis durch ein Terminal erfordert ein Terminal Certificate. Mit Hilfe eines gültigen Terminal Certificate kann dann z.B. auch im Rahmen der Web-Authentisierung auf einen elektronischen Personalausweis zugegriffen werden.

Terminal Certificates enthalten die Zugriffsberechtigungen des Terminals auf die Datenfelder eines ePA und spielen damit eine große Rolle in Bezug auf die Autorisation. Ein Terminal kann also nur solche Datenfelder auslesen, für die es anhand der Zugriffsberechtigungen befugt ist. Eine Änderung der Zugriffsberechtigungen ohne Anpassung der Signatur resultiert in dem Verlust der Gültigkeit des Terminal Certificate und damit zwangsläufig in einem Bruch der Vertrauenskette. Der ePA darf unter keinen Umständen ein Auslesen von Datenfeldern mit einem ungültigen Terminal Certificate zulassen. Dies kann durch entsprechende Gültigkeitsprüfungen im ePA sichergestellt werden.[9]

4.2.2 Schlüsselpaare

Jede Instanz der Public Key Infrastructure besitzt ein eigenes Schlüsselpaar. Hierunter sind mindestens die folgenden Instanzen zu verstehen:

- die Country Signing CA
- die Country Verifying CA
- jegliche Document Signer
- jegliche Document Verifier
- jegliche Ausweisdokumente (ePA)
- jegliche Terminals

Für die Beispiel-Implementierung wurden Schlüsselpaare entweder nach dem Digital Signature Algorithm (DSA) oder nach dem Diffie-Hellman-Verfahren (DH) in den Varianten für Elliptische Kurven-Kryptographie gewählt. Diese Verfahren werden im Folgenden kurz als ECDSA (Elliptic Curve Digital Signature Algorithm) bzw. ECDH (Elliptic Curve Diffie Hellman) bezeichnet.

4.2.2.1 Country Signing CA und Document Signer

Die Instanzen Country Signing CA und Document Signer besitzen ECDSA-Schlüsselpaare.

Für die Domain Parameter werden bei der Generierung der Schlüssel entsprechend den Anforderungen aus [ICAO-PKI-MRTD-ICC] folgende

9 Da der Ausweis keine eigene aktive Uhr besitzt, muss das aktuelle Datum z.B. anhand von vorhergehenden, erfolgreichen Verifikationen gesetzt werden. Details zur Lösung dieses Problems sind in diesem Dokument aus Platzgründen nicht enthalten, können aber z.B. in [BSI-TR-03110] Abschnitt 2.2.4 „Certificate Validation“ eingesehen werden.

Randbedingungen für den Signing Part eingehalten:

- minimum size for base point order for Country Signing CA keys: 256 bits
- minimum size for base point order for Document Signer keys: 224 bits
- minimum size for base point order for Terminal keys: 160 bits

4.2.2.2 Elektronischer Personalausweis

Der elektronische Personalausweis besitzt ECDH-Schlüsselpaare.
Für die Domain Parameter werden bei der Generierung der Schlüssel entsprechend den Anforderungen folgende Randbedingungen für den elektronischen Personalausweis eingehalten:

- Es gelten die Randbedingungen aus TR 03111 (vgl. [BSI-TR-03111]).
  Für die Beispiel-Implementierung sollten die Brainpool Domain Parameter aus [BrainpoolECC] verwendet werden.

4.2.2.3 Country Verifying CA, Document Verifier und Terminal Certificate

Die Instanzen Country Verifying CA, Document Verifier sowie Terminal besitzen ECDSA-Schlüsselpaare.
Für die Domain Parameter werden bei der Generierung der Schlüssel entsprechend den Anforderungen folgende Randbedingungen für den Verifier Part eingehalten:

- Es gelten die Randbedingungen aus TR 03111 (vgl. [BSI-TR-03111]).
  Für die Beispiel-Implementierung sollten die Brainpool Domain Parameter aus [BrainpoolECC] verwendet werden.
- Ferner gilt nach [BSI-TR-03110] D.3.3, dass Domain Parameter ausschließlich in CVCA Certificates enthalten sein dürfen. Die Domain Parameter für Document Verifier und Terminal dürfen daher nicht gesetzt sein. Die Domain Parameter für DV Certificates und Terminal Certificates werden von der darüber liegenden CVCA vererbt:

> „In CV certificates the optional domain parameters MUST be either all present or all absent. Only CVCA Link Certificates MAY contain domain parameters. The domain parameters for DV and terminal public keys are inherited from the respective CVCA public key and MUST be absent.“ (D.3.3 [BSI-TR-03110])

4.2.3 Abläufe

Im Rahmen der PKI existieren die folgenden Abläufe:

- Erzeugen einer Country Verifying CA
- Erzeugen und Signieren eines Country Verifying CA Certificates
- Erzeugen eines Document Verifiers
- Erzeugen und Signieren eines Document Verifier Certificates
- Erzeugen eines Terminals
- Erzeugen und Signieren eines Terminal Certificates
- Erzeugen einer Country Signing CA
- Erzeugen und Signieren eines Country Signing CA Certificates
- Erzeugen eines Document Signers
- Erzeugen und Signieren eines Document Signer Certificates
- Erzeugen eines elektronischen Personalausweises
- Signieren eines elektronischen Personalausweises

Diese o.g. Abläufe werden im Detail im Kapitel 5 (Ablaufbeschreibung) ab Seite 64 beschrieben.

4.3 Der elektronische Personalausweis (ePA)

Der elektronische Personalausweis ist ein Chip mit SmartCard-Funktionalität.

Er muss die folgenden Schlüsselpaare enthalten oder während des Ablaufs generieren können:

- ein statisches Diffie-Hellman-Schlüsselpaar (static ECDH-Keypair)
- ein flüchtiges Diffie-Hellman-Schlüsselpaar (ephemeral ECDH-Keypair)

Das statische ECDH-Schlüsselpaar des ePA wird im Folgenden als **`keyPairPiccEcdhEpa`** bezeichnet. Es besteht aus dem öffentlichen Schlüssel **`pkPicc`** bzw. **`PK_PICC`** und dem geheimen Schlüssel **`skPicc`** bzw. **`SK_PICC`**.

Das flüchtige ECDH-Schlüsselpaar des ePA wird im Folgenden als **`ephemeralKeyPairPiccEcdhEpa`** bezeichnet. Es besteht aus dem öffentlichen Schlüssel **`ephemeralPkPicc`** bzw. **`~PK_PICC`** und dem geheimen Schlüssel **`ephemeralSkPicc`** bzw. **`~SK_PICC`**.

Schlüsselpaar	Bestandteile	Typ
`keyPairPiccEcdhEpa`	`(pkPicc, skPicc) bzw. (PK_PICC, SK_PICC)`	ECDH
`ephemeralKeyPairPiccEcdhEpa`	`(ephemeralPkPicc, ephemeralSkPicc) bzw. (~PK_PICC, ~SK_PICC)`	ECDH

Tabelle 4: Schlüsselpaare des ePA

4.3.1 Erzeugen des statischen ECDH-Schlüsselpaars

Das statische ECDH-Schlüsselpaar **`keyPairPiccEcdhEpa`** wird bei der Herstellung des ePA erzeugt und in dem ePA gespeichert. Hierbei wird der öffentliche Schlüssel **`pkPicc`** in signierter Form in einem Read-only-Bereich hinterlegt. Der öffentliche Schlüssel **`pkPicc`** wird deswegen signiert, damit Kommunikationspartner (z.B. der EAS) später verifizieren können, ob es sich um einen authentischen öffentlichen Schlüssel handelt.
Der geheime Schlüssel **`skPicc`** wird in sicherem Speicher abgelegt, der von Dritten weder ausgelesen noch beschrieben werden kann.

Für die Beispiel-Implementierung wird aufgrund des objekt-orientierten Ansatzes die Generierung des statischen Schlüsselpaares als eine Funktionalität des elektronischen Personalausweises realisiert. Für eine detaillierte Beschreibung des Ablaufs der Erstellung sowie der Personalisierung des elektronischen Personalausweises wird auf Kapitel 5.1.5 (Erstellung und Personalisierung des ePA) verwiesen.

Die Domain Parameter bei der Erzeugung des Schlüsselpaars müssen der BSI TR 03111 genügen. Die Spezifikation empfiehlt für ECC die sog. Brainpool Domain Parameter (siehe [BSI-TR-03111]). Für die Beispiel-Implementierung wurden die Brainpool Domain Parameter **`brainpoolP224r1`** aus [BrainpoolECC] gewählt.

4.3.2 Erzeugen des flüchtigen ECDH-Schlüsselpaars

Das flüchtige ECDH-Schlüsselpaar **`ephemeralKeyPairPiccEcdhEpa`** wird im Rahmen des Password Authenticated Connection Establishment (PACE)

erzeugt.

> **Bemerkung**
> Zum Zeitpunkt der Anfertigung dieser Arbeit existierte keine detaillierte Spezifikation des elektronischen Personalausweises. Ferner war kein Prototyp verfügbar, sodass in einer Beispiel-Implementierung diese Komponente vorerst nur durch eine Softwaresimulation realisiert werden kann.

4.3.3 Schnittstellen

Der ePA kommuniziert indirekt mit dem eID Authentication Server über den RF-Leser, das Applet und den eID Authentication Proxy (EAP) (siehe auch Abbildung 10). Diese Schnittstelle wird als Schnittstelle ePA – Applet bezeichnet. Die Kommunikation auf der Strecke zwischen dem ePA und dem RF-Leser wird über ISO 14443 abgewickelt. Der Treiber für den RF-Leser wird in der Regel Kommandos nach ISO/IEC 7816-4 akzeptieren ([ISO/IEC 7816-4], [BSI-TR-03110]). Das Applet nutzt die API des Treibers für den RF-Leser.

4.3.3.1 Schnittstelle ePA – Applet

Die Schnittstelle zwischen dem ePA und dem Applet ist in der Regel eine lokale Schnittstelle, d.h. die Kommunikation über diese Schnittstelle findet nicht über ein Netzwerk statt.
Das Applet kommuniziert mit dem RF-Leser. Der RF-Leser ist ein transparentes Element und kann daher in der Betrachtung der Kommunikation vernachlässigt werden. Der RF-Leser leitet lediglich die Nachrichten des Applet an den ePA weiter und gibt die Antwortnachrichten des ePA an das Applet zurück.
Der ePA muss seine Nachrichten an den EAS adressieren. Nach erfolgreichem Abschluss der SETUP-Phase (siehe Abschnitt 5.3) werden an der Schnittstelle zwischen ePA und Applet keine Informationen im Klartext übertragen.

Das Format der Nachrichten dieser Schnittstelle hängt in hohem Maße davon ab, in welchem Format die Treiber der RF-Lesegeräte Nachrichten erwarten. Zu diesem Aspekt kann derzeit noch keine nähere Aussage gemacht werden. Es ist wünschenswert, dass der Treiber eine objekt-orientierte Schnittstelle zum

RF-Leser bietet.

4.4 Der RF-Leser

Der RF-Leser ist die Gegenstelle des ePA für die Funkkommunikation. Er leitet die Nachrichten des Applet an den ePA weiter und umgekehrt. Er verändert den Inhalt der Nachrichten nicht, sondern dient lediglich als Vermittlungseinheit.

Aufgrund der Tatsache, dass der elektronische Personalausweis zum aktuellen Stand lediglich als Softwaresimulation realisiert wird, ergibt sich für die RF-Leser-Komponente, dass diese ebenfalls als Softwaresimulation realisiert wird.

Der RF-Leser wird in der Kommunikation zwischen ePA und dem Applet als transparentes Element angesehen. Für weitere Informationen und die Schnittstellenbeschreibung wird auf Abschnitt 4.3.3.1 (Schnittstelle ePA – Applet) ab Seite 48 verwiesen.

4.5 Das Applet

Das Applet ist für die Interaktion des Benutzers mit dem Authentikations-mechanismus im ePA zuständig. Darüber hinaus koordiniert es die Kommunikation zwischen dem ePA und den Server-Komponenten. Für die Kommunikation über das Internet ist das Applet die Gegenstelle des Servers.

Das Applet repräsentiert die Software, die auf dem Client-PC läuft und mit dem RF-Leser kommunizieren kann. Darüber hinaus initiiert das Applet die Verbindungen zum Server. Dies ist sinnvoll, da Clients sich häufig hinter Firewalls geschützt befinden und lediglich Verbindungen vom Client zum Server aufgebaut werden können. Umgekehrt ist dies in der Regel durch eine Firewall unterbunden oder aufgrund von Network Address Translation (NAT, [RFC 2663]) nur schwer möglich.[10]

10 Sog. NAT traversal Techniken (wie beispielsweise UDP hole punching [Schmidt 2006]) können dazu benutzt werden, um bidirektionale Kommunikation von Hosts im Internet mit Rechnern hinter einem NAT-Gateway aufzubauen. Diese Techniken sind vielfältig, aber es gibt keinen eindeutigen, verlässlichen Mechanismus.

Das Applet beinhaltet kaum kryptographische Funktionalität. Es handelt sich bei dem Applet in Bezug auf die Kommunikation lediglich um eine Weiterleitung von Nachrichten (Proxy). Gegebenenfalls werden die Nachrichten umformatiert, die Nutzdaten der Nachrichten werden dabei jedoch nicht verändert.

Die Aufgaben des Applets gliedern sich in die folgenden Bereiche:

- Interaktion mit dem Benutzer
- Nachrichtenvermittlung
- Ablaufkontrolle

4.5.1 Interaktion mit dem Benutzer

Das Applet ist die Komponenten, die mit dem Benutzer interagiert. Es wird von dem Webserver geladen, mit dem die Web-Authentisierung stattfinden soll.

Zu Beginn erfragt das Applet die PIN des elektronischen Personalausweises, die im Rahmen des PACE benötigt wird.
Darüber hinaus informiert das Applet den Benutzer permanent über den Ablauf der Authentisierung. Hierbei empfiehlt sich die Realisierung in Form einer Aktivitätsanzeige (activity indicator) und/oder einer Fortschrittsanzeige (progress indicator).
Sollte es während des Ablaufs zu einem Fehler kommen, muss das Applet den Benutzer hierüber entsprechend informieren.
Wenn die Kommunikation erfolgreich abgeschlossen wurde, kann das Applet dies dem Benutzer signalisieren. Darüber hinaus kann das Applet wahlweise auch den Browser dazu veranlassen, einen Seitenwechsel vorzunehmen, um nach erfolgreicher Authentisierung beispielsweise den elektronischen Bestellvorgang fortzusetzen.

4.5.2 Nachrichtenvermittlung

Das Applet ist dafür zuständig, Anforderungen an den eID Authentication Server zu schicken und die Antworten entgegen zu nehmen. Basierend auf den Datensätzen der Antworten des Servers werden die Nachrichten für den elektronischen Personalausweis aufbereitet. Diese gegebenenfalls konvertierten Nachrichten werden anschließend an den ePA übergeben.

Das Applet agiert also lediglich als Vermittlungsinstanz für Nachrichten. Dies ist sinnvoll, da die kryptographische Arbeit ausschließlich in den Endpunkten, also im eID Authentication Server sowie im elektronischen Personalausweis stattfindet. Als Folge dessen handelt es sich beim Applet um eine eher unkritische Komponente, die selbst bei gezielter Manipulation nicht zum Bruch der Kryptographie verwendet werden kann, sondern lediglich für Denial-of-Service-Zwecke. Details hierzu werden in Kapitel 4.5.5 (Vertrauenswürdigkeit des Applets) ab Seite 53 behandelt.
Sobald der sichere Kanal zwischen dem eID Authentication Server und dem elektronischen Personalausweis aufgebaut ist, kann das Applet den Inhalt der Nachrichten nicht mehr einsehen, da die Nachrichten von den Endpunkten verschlüsselt und vor Manipulation durch Anwendung eines Message Authentication Code geschützt werden. Das Auslesen der personenbezogenen Daten findet lediglich über diesen sicheren Kanal statt. Das Applet bekommt daher keine Kenntnis von den personenbezogenen Daten.

4.5.3 Ablaufkontrolle

Aufgabe des Applet ist darüber hinaus die Kontrolle des korrekten Ablaufs. Dazu beinhaltet es einen Zustandsautomaten, der die korrekte Reihenfolge von Nachrichten ermöglicht und überwacht. Wenn Nachrichten in einer falschen Reihenfolge oder in einem falschen Kontext eintreffen, beendet das Applet die Kommunikation und leitet gegebenenfalls eine Re-Initialisierung ein. Dies ist eine Möglichkeit, um ein unerwünschtes Auslesen von personenbezogenen Daten zu verhindern.[11]

4.5.4 Screendesign der Beispiel-Implementierung

Im Rahmen der Beispiel-Implementierung besitzt das Applet drei Screendesigns. Diese Designs werden hier vorgestellt, um wichtige Aspekte bei der Gestaltung hervorzuheben.
Zunächst muss das Applet die PIN des Benutzers erfragen. Hierzu wurde das folgende Screendesign entwickelt (siehe Abbildung 15).

11 Die Anwendung dieses Mechanismus alleine bietet noch keinen ausreichenden Schutz vor unerwünschtem Zugriff auf Daten. Zusätzlich müssen die Endpunkte, d.h. der eID Authentication Server sowie der elektronische Personalausweis einen derartigen Schutz aufweisen. Dies kann beispielsweise ebenfalls durch Zustandsautomaten in den entsprechenden Komponenten erfolgen.

Abbildung 15: Screendesign des Applets im ersten Schritt "Abfrage der PIN"

Das Applet zeigt dem Benutzer in der Überschrift den Namen des Prozesses an, in diesem Fall ist es die eID Web-Authentisierung. Ferner wird der Benutzer aufgefordert, seine PIN in das Eingabefeld einzugeben. Er bestätigt die Eingabe durch Drücken des Knopfes mit der Beschriftung „Los“ oder durch Drücken der Taste „ENTER“. Der Umriss des Applets wird dem Benutzer durch einen Rahmen (in Abbildung 15 gestrichelt) vermittelt.

Nach der Eingabe der PIN wechselt das Screendesign in die In-Process-View (siehe Abbildung 16). Hierbei steht der Prozessfortschritt im Vordergrund.

Abbildung 16: Screendesign des Applets im zweiten Schritt "eID Authentisierung"

Eine Aktivitätsanzeige zeigt an, wenn das Applet aktiv ist. Eine Fortschrittsanzeige visualisiert den Fortschritt im Authentisierungsprozess.
Im Erfolgsfall benötigt der Benutzer keine weiteren Informationen. Er kann sich jedoch über einen Button ein Log-Fenster anzeigen lassen, indem die Abfolge

protokolliert wird. Darüber hinaus sollte es eine Möglichkeit geben, mit Hilfe derer der Benutzer Informationen zum auslesenden Terminal erhält. Dies umfasst die Darstellung der Informationen aus dem Terminal Certificate. Das Screendesign hierzu ist in Abbildung 17 dargestellt.

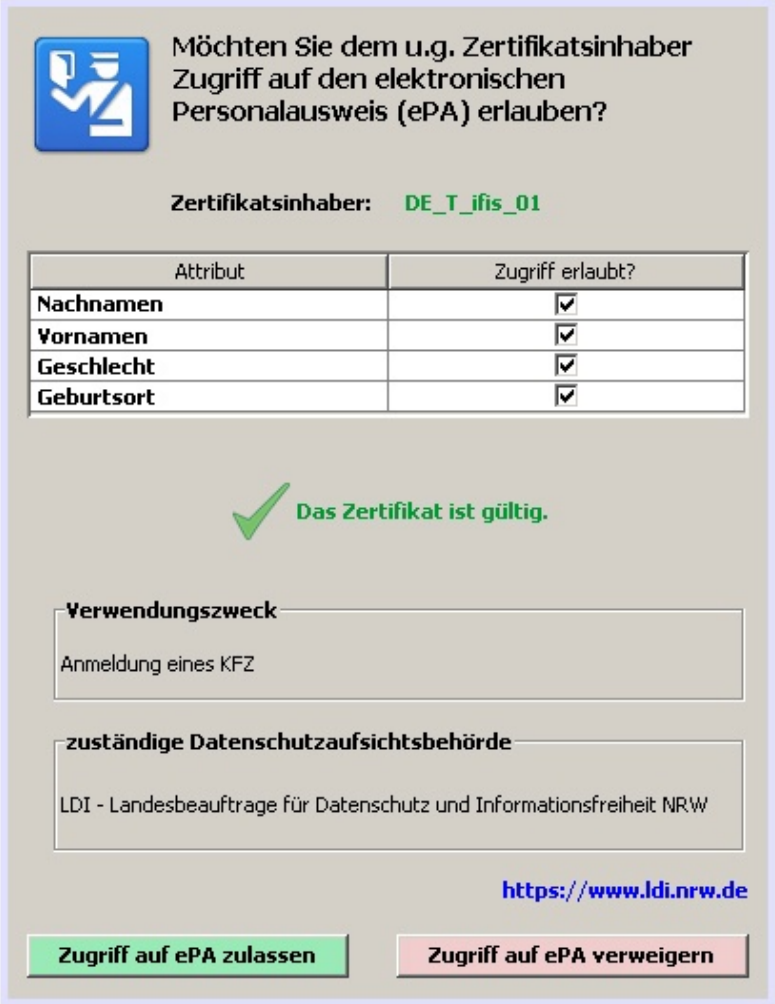


Abbildung 17: Screendesign des Terminal Certificate Informationsdialogs

4.5.5 Vertrauenswürdigkeit des Applets

Grundsätzlich könnte sich folgendes Sicherheitsproblem ergeben: Wenn die auf dem Client eingesetzte Software, die den RF-Leser und damit den Chip ansteuert – z.B. das Applet – nicht vertrauenswürdig ist, könnte möglicherweise

die Gefahr des Missbrauchs (z.B. Man-in-the-middle Angriff) bestehen. Um dieses Problem zu lösen, wird im Rahmen der Beispiel-Implementierung festgelegt, dass das Applet signiert werden muss. Idealerweise wäre eine Signatur des Applets durch einen Document Verifier (DV) wünschenswert, die sich bei der Verifikation auf den öffentlichen Schlüssel der CVCA im ePA abstützt. Auf diese Weise könnte mit höherer Wahrscheinlichkeit verhindert werden, dass Programme (und damit auch Applets) zur Ausführung kommen, die nicht von der CVCA (z.B. im Namen der Bundesrepublik Deutschland) als vertrauenswürdig erachtet wurden. Dies ist jedoch zum aktuellen Zeitpunkt nicht möglich. Der folgende Abschnitt erläutert den Grund aus technischer Perspektive.[12]
Das Java-Plugin übernimmt die Verifikation der Signatur eines Applets. Hierzu ist das Java-Plugin, ähnlich wie bei einem Browser, mit Root-Zertifikaten von vertrauenswürdigen Instanzen ausgestattet. Selbstverständlich können hier weitere vertrauenswürdige Wurzeln, wie beispielsweise ein Root-Zertifikat der CVCA hinzugefügt werden. Es kann jedoch nicht forciert werden, von welcher Root-CA ein Applet signiert worden sein muss. Ein Angreifer könnte sich also ein möglicherweise bösartiges Applet von einer Root-CA seiner Wahl, die nicht notwendigerweise die CVCA ist, signieren lassen. Unabhängig davon, ob sich eine Root-CA finden ließe, die das bösartige Applet signiert, ist für die Sicherheit entscheidend, dass nicht garantiert werden kann, dass die Verifikation das Applet für vertrauenswürdig erklärt.

Trotz der oben genannten Einschränkungen bietet die Signatur des Applets einen erheblichen Schutz. Die in der Regel von jeder CA geprüften und durch eine Signatur geschützten Merkmale eines Applets sind:

- der Herausgeber
- die zertifizierende Instanz
- der Name des Applets

Das wichtigste verlässliche Merkmal, das im Rahmen des Signatur-Verfahrens von Code interessiert, ist der Herausgeber. Die Identität des Herausgebers muss üblicherweise durch amtliche Dokumente belegt werden. Ein Angreifer könnte also das Applet zwar signieren lassen, jedoch nicht im Namen einer

12 Abgesehen von der technischen Realisierbarkeit würde dieses Code Signing bzw. der zwangsläufig vorhergehende Code Review einen erheblichen Aufwand für die CVCA bedeuten.

dritten Terminal-Instanz (als Herausgeber). Auf dieses Verfahren stützen sich im Übrigen auch die Applets der Online-Steuerverwaltung ELSTER ([ELSTER 2008]) ab.

Abbildung 18: Beispiel für einen Dialog eines erfolgreich verifizierten Java-Applets (in diesem Fall das elsteranalyze-Applet von [ELSTER 2008])

Es ist wichtig, hervorzuheben, dass ein nicht-vertrauenswürdiges Applet die Sicherheit des Authentisierungsprozesses nicht maßgeblich gefährdet. Mögliche Angriffe und deren Auswirkungen auf den Authentisierungsprozess werden in den Abschnitten Fehler: Referenz nicht gefunden und fortfolgend beschrieben.

4.5.6 Schnittstellen

Das Applet hat die folgenden Schnittstellen:

- Schnittstelle Applet – EAP
- Schnittstelle Applet – RF-Leser

Die Schnittstelle zwischen Applet und EAP wird im Abschnitt 4.6.1.1 (Schnittstelle EAP – Applet) ab Seite 57 beschrieben.

Die Schnittstelle zwischen Applet und RF-Leser bzw. ePA wird in Abschnitt 4.3.3.1 (Schnittstelle ePA – Applet) ab Seite 48 beschrieben.

4.6 Der eID Authentication Proxy (EAP)

Der eID Authentication Proxy ist eine serverseitige Komponente. Er erweitert

die Web-Portal-Software um die notwendige Funktionalität, damit die durch den eID Authentication Server gewonnenen, personenbezogenen Daten angemessen verarbeitet werden können.

Die Aufgaben des eID Authentication Proxy sind:

- Nachrichtenvermittlung zwischen EAS und dem Applet
- Session-Kontrolle
- möglicherweise die Verarbeitung der personenbezogenen Daten

In erster Linie agiert der eID Authentication Proxy als Weiterleitungsinstanz von Nachrichten, die zwischen EAS und dem Applet übermittelt werden. Dabei kontrolliert es, dass Nachrichten nur in der gültigen Reihenfolge übertragen werden (Session-Kontrolle) und unterbindet oder reinitialisiert die Kommunikation im Fehlerfall. Sobald die Übertragung der Attribute abgeschlossen ist, werden diese Daten oder ein Teil davon möglicherweise an den EAP gegeben, damit dieser sie im weiteren Verlauf verarbeitet. Diese Verarbeitung könnte beispielsweise die Anzeige der Daten vor dem Abschluss einer Bestellung in einem Web-Shop sein.
Idealerweise sollten die übermittelten Attribute den EAS nicht verlassen. Sollten die Daten den EAS nach Abschluss der Transaktion verlassen und beispielsweise über eine unverschlüsselte Verbindung zum Applet geschickt werden (z.B. zur Darstellung), so könnte ein Angreifer die Daten hier abfangen. Die Daten sollten – wenn auf eine Übertragung zum Client nicht verzichtet werden kann – ausschließlich über einen verschlüsselten Kanal (z.B. HTTPS) übermittelt werden.

4.6.1 Schnittstellen

Der eID Authentication Proxy hat die folgenden Schnittstellen:

- Schnittstelle EAP – EAS
- Schnittstelle EAP – Applet

Die Schnittstelle zwischen EAP und EAS wird im Abschnitt 4.7.5.1 (Schnittstelle EAS – EAP) ab Seite 62 behandelt.
Die Schnittstelle zwischen EAP und dem Applet wird im folgenden Abschnitt beschrieben.

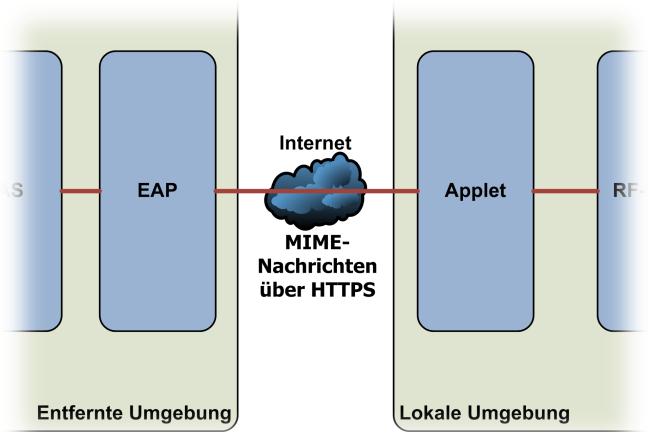


Abbildung 19: Ausschnitt aus Abbildung 10, Schnittstelle zwischen EAP und Applet im Detail

4.6.1.1 Schnittstelle EAP – Applet

Das Applet kommuniziert mit dem EAP über das Internet (siehe Abbildung 19). Da es sich bei der Web-Authentisierung um eine Applikation des World Wide Web handelt, liegt es nahe, dass für die Kommunikation über diese Schnittstelle das Hypertext Transfer Protocol (HTTP, [RFC 2616]) genutzt werden kann. Es ist zu empfehlen, dass die HTTP-Kommunikation per Transport Layer Security (TLS bzw. SSL) abgesichert wird (HTTPS, [RFC 2818], [RFC 2817]).

Eine mögliche Variante die Kommunikation über die Schnittstelle abzuwickeln besteht darin, sog. MIME-Nachrichten einzusetzen. Multipurpose Internet Mail Extensions (MIME, [RFC 2045], [RFC 2046], [RFC 2047], [RFC 2049], [RFC 2387], [RFC 4289], [RFC 4288]) ist ein Standard für die Kodierung von Internetnachrichten für diverse Zwecke. Ursprünglich wurde MIME lediglich für E-Mail entworfen. Heute wird daher ein Großteil aller E-Mails unter der Verwendung von MIME übertragen. MIME findet heutzutage jedoch auch in anderen Internet-Applikationen Anwendung, so unter anderem bei der Übertragung von Formularinhalten im Web oder bei der Beschreibung von Dateiformaten.
MIME kapselt beliebige Inhalte. Hierzu werden die zu übertragenden Daten bei Bedarf auf der Seite des Absenders kodiert und beim Empfänger dekodiert.

Dies wird auch als Transfer-Encoding bezeichnet. So wird erreicht, dass binäre Daten selbst auf rein textbasierten Übertragungswegen einwandfrei übermittelt werden können.

Die Nachrichten der Schnittstelle zwischen EAP und dem Applet stellen MIME-Nachrichten, die aus mehreren Teilen bestehen, sog. Multipart Messages, dar. Dabei wird jeder zu übertragende Parameter in einem eigenen Teil, sog. MIME-Part, übertragen.

4.7 Der eID Authentication Server (EAS)

Der eID Authentication Server (EAS) ist eine Serverkomponente. Er entspricht dem sog. Remote Terminal und bildet damit einen Endpunkt in der Kommunikation. Der EAS ist die Gegenstelle des Secure Messaging beim Auslesen der Datenfelder des ePA. Dies bedeutet, dass im Rahmen der sicheren Kommunikation ausschließlich der EAS die personenbezogenen Daten entschlüsseln kann.

Aufgabe des EAS ist die Kommunikation mit dem ePA, insbesondere das authentifizierte Auslesen der Datenfelder des ePA. Darüber hinaus muss er die personenbezogenen Daten nach der Akquise an die weiterverarbeitende Komponente, z.B. die Webportal-Software, weitergeben.
Hierzu beinhaltet der EAS die kryptographische Funktionalität, die zur Kommunikation mit dem ePA über einen sicheren Kanal benötigt wird.

4.7.1 Schlüsselpaare

Der EAS muss die folgenden Schlüsselpaare enthalten oder während des Ablaufs generieren können:

- ein statisches Elliptic-Curve-DSA-Schlüsselpaar (static ECDSA-Keypair)
- ein flüchtiges Diffie-Hellman-Schlüsselpaar (ephemeral ECDH-Keypair)

Das statische ECDSA-Schlüsselpaar wird im Folgenden als **`keyPairPcdEcdsaEas`** bezeichnet. Es besteht aus dem öffentlichen Schlüssel **`pkPcd`** bzw. **`PK_PCD`** und dem geheimen Schlüssel **`skPcd`** bzw. **`SK_PCD`**.

Das flüchtige ECDH-Schlüsselpaar wird im Folgenden als **ephemeralKeyPairPcdEcdhEas** bezeichnet. Es besteht aus dem öffentlichen Schlüssel **ephemeralPkPicc** bzw. **~PK_PICC** und dem geheimen Schlüssel **ephemeralSkPicc** bzw. **~SK_PICC**.

Schlüsselpaar	Bestandteile	Typ
keyPairPcdEcdsaEas	(pkPcd, skPcd) bzw. (PK_PCD, SK_PCD)	ECDSA
ephemeralKeyPairPcdEcdhEas	(ephemeralPkPcd, ephemeralSkPcd) bzw. (~PK_PCD, ~SK_PCD)	ECDH

Tabelle 5: Schlüsselpaare des eID Authentication Servers (EAS)

Der EAS benötigt ein Terminal Certificate[13], um die Daten des ePA auslesen zu können. Das Terminal Certificate enthält den öffentlichen Schlüssel **PK_PCD**. Es muss von einem Document Verifier signiert sein.

4.7.2 Erzeugen des statischen ECDSA-Schlüsselpaars

Das statische ECDSA-Schlüsselpaar **keyPairPcdEcdsaEas** wird einmalig bei der Erzeugung des Terminals erzeugt und idealerweise in einem (Hoch-)Sicherheitsmodul gespeichert. Hierbei wird der öffentliche Schlüssel **pkPcd** in Form eines Terminal Certificates hinterlegt. Der öffentliche Schlüssel **pkPcd** wird in Form des Terminal Certificates dabei von einem Document Verifier signiert. Somit können Kommunikationspartner (z.B. der ePA) später verifizieren, ob es sich um einen authentischen öffentlichen Schlüssel und damit um ein authentisches Terminal handelt.
Der geheime Schlüssel **skPcd** wird in sicherem Speicher z.B. in einem (Hoch-)Sicherheitsmodul abgelegt.

Für die Beispiel-Implementierung wird aufgrund des objekt-orientierten Ansatzes die Generierung des statischen Schlüsselpaares als eine Funktionalität des eID Authentication Servers (EAS) realisiert. Für eine detaillierte Beschreibung des Ablaufs der Erstellung des eID Authentication Servers siehe Kapitel 6.5 (eID Authentication Server (EAS)) ab Seite 90.

13 Siehe Abschnitt 4.7.4

Die Domain Parameter bei der Erzeugung des Schlüsselpaars müssen der BSI TR 03111 genügen. Die Spezifikation empfiehlt für ECC die sog. Brainpool Domain Parameter (siehe [BSI-TR-03111]). Für die Beispiel-Implementierung wurden die Brainpool Domain Parameter **`brainpoolP224r1`** aus [BrainpoolECC] gewählt.

> **Bemerkung**
> Es ist zu beachten, dass die BSI TR 03110 vorschreibt, dass die Domain Parameter zur Generierung eines Schlüsselpaars eines Terminals nicht frei gewählt werden dürfen, sondern abhängig sind von den Domain Parametern, die die darüber liegenden Instanzen Document Verifier sowie in erster Linie Country Verifying CA vorschreiben:

„The domain parameters for DV and terminal public keys are inherited from the respective CVCA public key [and MUST be absent].“ ([BSI-TR-03110], D.3.3, p. 68)

> Dies muss bereits bei der Generierung der statischen Schlüssel beachtet werden. Hierzu bietet es sich an, dass die Country Verifying CA die Terminals (und die Document Verifier) über die zu benutzenden Domain Parameter informiert.

4.7.3 Erzeugen des flüchtigen ECDH-Schlüsselpaars

Das flüchtige ECDH-Schlüsselpaar **`ephemeralKeyPairPcdEcdhEas`** wird im Rahmen der Terminal Authentication erzeugt. Für die Erzeugung des Schlüsselpaars müssen die vom elektronischen Personalausweis übermittelten Domain Parameter genutzt werden.
Der Ablauf der Authentisierung wird im Detail in Abschnitt 5.3 (Die SETUP-Phase) beschrieben.

4.7.4 Terminal Certificate (TC)

Das Terminal Certificate ist ein wichtiger Bestandteil der Authentisierung zwischen ePA und EAS. Das Terminal Certificate enthält neben dem

öffentlichen Schlüssel des EAS die Berechtigung des EAS. Die Berechtigung gibt an, welche Felder des ePA durch den EAS ausgelesen werden dürfen. Das Terminal Certificate muss durch einen gültigen Document Verifier signiert worden sein. Auf diese Weise kann ein ePA sicherstellen, dass nur die Inhalte der Datenfelder an den EAS gesendet werden müssen, für die der EAS auch tatsächlich authentifiziert ist.
Die Berechtigungen des Terminal Certificate werden dabei entsprechend der Beschreibung in Abschnitt C.4 sowie der Tabelle C.4 aus [BSI-TR-03110] kodiert. Grundsätzlich bestimmen sich die Berechtigungen danach, um welche Art von Terminal es sich handelt. Dies wird in der Role ID angegeben. Für die eID-Anwendung handelt es sich dabei um den Object Identifier `0.4.0.127.0.7.3.1.2.2`, der sich wie folgt zusammen setzt:

OID/OID-Part	Bedeutung
`0.4.0.127.0.7`	bsi-de
`.3`	.applications
`.1`	.mrtd
`.2`	.id-roles
`.2`	.eID application

Tabelle 6: Object Identifier für die eID-Anwendung im Rahmen der Certificate Holder Authorization

Die sog. relative Authorization, in der die Zugriffsberechtigungen auf die einzelnen Data Groups kodiert sind, wird als Bitmuster in 5 Bytes anhand der folgenden Tabelle kodiert.

39	38	37	...	33	32	...	29	28	...	8	7	...	3	2	1	0	Description
x	x	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	**Role**
1	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	CVCA
1	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	DV (official domestic)
0	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	DV (commercial, other)
0	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Terminal
-	-	x	x	x	-	-	-	-	-	-	-	-	-	-	-	-	**Write Access**
-	-	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	EF.17
-	-	-	⋱	-	-	-	-	-	-	-	-	-	-	-	-	-	⋮
-	-	-	-	1	-	-	-	-	-	-	-	-	-	-	-	-	EF.21
-	-	-	-	-	x	x	x	-	-	-	-	-	-	-	-	-	**RFU: R/W Access**
-	-	-	-	-	-	-	-	x	x	x	-	-	-	-	-	-	**Read Access**
-	-	-	-	-	-	-	-	1	-	-	-	-	-	-	-	-	EF.21
-	-	-	-	-	-	-	-	-	⋱	-	-	-	-	-	-	-	⋮
-	-	-	-	-	-	-	-	-	-	1	-	-	-	-	-	-	EF.1
-	-	-	-	-	-	-	-	-	-	-	x	x	x	x	x	x	**Special Functions**
-	-	-	-	-	-	-	-	-	-	-	1	-	-	-	-	-	eID-PIN waived
-	-	-	-	-	-	-	-	-	-	-	-	0	-	-	-	-	RFU
-	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-	-	Unblock eID-PIN
-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-	Restricted Identification
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-	Install eSign
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	Age Verification

Abbildung 20: Table C.4 aus [BSI-TR-03110] Kodierung der relative authorization der Certificate Holder Authorization

Ein Beispiel für ein CVCA-Zertifikat ist in Abbildung 14 auf Seite 43 zu sehen.

4.7.5 Schnittstellen

Der EAS hat eine Schnittstelle zum eID Authentication Proxy (EAP), die im folgenden Abschnitt beschrieben wird.

4.7.5.1 Schnittstelle EAS – EAP

Bei der Schnittstelle zwischen EAP und EAS handelt es sich um eine webbasierte Schnittstelle. Der EAP ist in der Regel eine Komponente eines Content Management Systems. Darüber hinaus nimmt der EAP in seiner Kommunikation mit dem Applet MIME-Nachrichten entgegen. Daher liegt es nahe, auch in der Kommunikation zwischen EAP und EAS MIME-Nachrichten zu verwenden.

Neben den Nutzdaten, die zwischen dem ePA und dem EAS ausgetauscht

werden, sollte der EAS den EAP mit Zustandsinformationen versorgen. Spätestens nach dem erfolgreichem Aufbau eines sicheren Kanals zwischen dem ePA und dem EAS kann keine dazwischenliegende Komponente – wie u.a. der EAP – den Inhalt der Nachrichten abgreifen. Damit der EAP entscheiden kann, ob und wann die Verbindung zwischen EAP und Applet abgebaut werden soll, muss der EAS den EAP mit Zustandsinformationen versorgen. Die Schnittstelle zwischen EAS und EAP erfordert also die Übertragung der folgenden Informationen:

- Informationen des Protokolls zwischen EAS und ePA (Nutzdaten)
- Zustandsinformationen aus dem EAS

Die Nutzdaten werden als MIME-Parts vom EAP an den EAS weitergeleitet. In umgekehrter Richtung werden ebenfalls MIME-Parts verwendet. Pro Datum wird dabei ein MIME-Part verwendet.

Die Zustandsinformationen können als HTTP-Header oder als zusätzliche MIME-Parts übertragen werden. Die Zustandsinformationen sollten jedoch vom EAP herausgefiltert werden, sodass sie nicht an das Applet weitergeleitet werden. Ein möglicher Zustandsautomat der Schnittstelle zwischen EAS und EAP sieht dabei wie folgt aus.

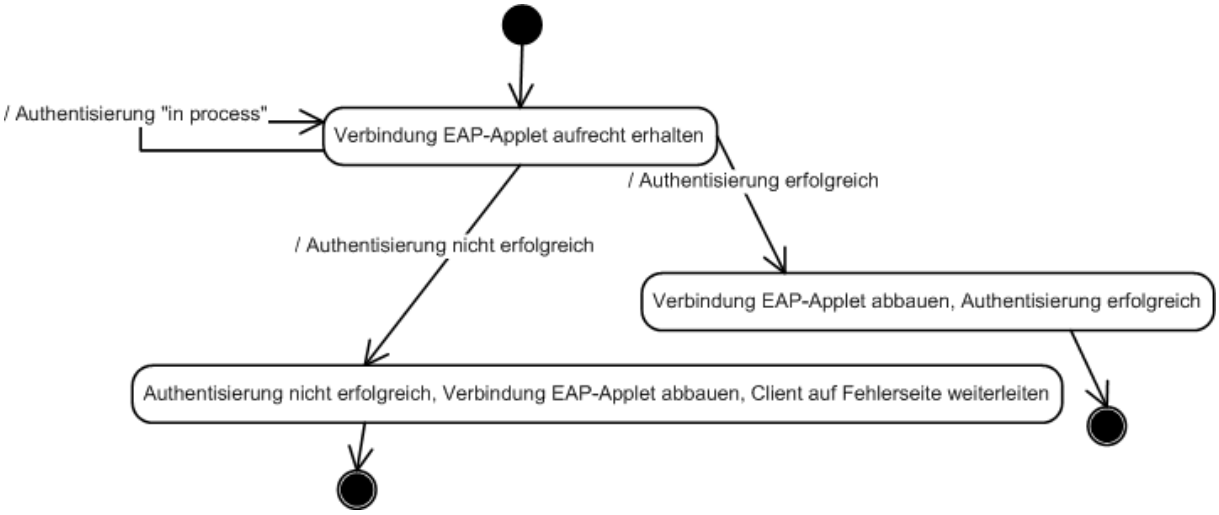


Abbildung 21: Zustandsautomat der Schnittstelle zwischen EAS und EAP

5 Ablaufbeschreibung

Dieses Kapitel enthält die Beschreibungen der wichtigsten Abläufe, die an den Komponenten auftreten.

5.1 PKI-Abläufe

5.1.1 Abläufe einer Country Verifying CA

Eine Country Verifying CA wird wie folgt erzeugt:

- Generieren eines ECDSA-Schlüsselpaars unter Beachtung der Bedingungen aus 4.2.2.3 (Country Verifying CA, Document Verifier und Terminal Certificate) ab Seite 45
- Erzeugen eines Zertifikats der CVCA als Card Verifiable Certificate (CVC)
- Signieren des Zertifikats (Selbstsignatur)
- Verteilen des Zertifikats

Im Rahmen der Verteilung muss das CVCA-Zertifikat auf sicherem Weg in den ePA gelangen. Der ePA nutzt das CVCA-Zertifikat, um die Echtheit von Terminals zu überprüfen.

Die Verteilung des CVCA-Zertifikats ist ein sicherheitskritischer Prozess. Wenn ein gefälschtes CVCA-Zertifikat ausgerollt wurde, können die Lesegeräte nicht verlässlich die Authentizität der Ausweisdokumente prüfen.[14]

Darüber hinaus muss eine CVCA Document Verifier zulassen und signieren.

5.1.2 Abläufe eines Document Verifiers

Ein Document Verifier wird wie folgt erzeugt:

- Generieren eines ECDSA-Schlüsselpaars unter Beachtung der Bedingungen aus 4.2.2.3 (Country Verifying CA, Document Verifier und Terminal Certificate) ab Seite 45
- Erzeugen eines Zertifikats des DV als Card Verifiable Certificate (CVC)
- Signieren des Zertifikats durch die Country Verifying CA
- Das Zertifikat des Document Verifier wird an die Terminals weitergegeben,

14 Interessanterweise können auf der Webseite der belgischen Root CA die Zertifikate nicht über eine SSL-geschützte Verbindung abgerufen werden, sondern nur über unverschlüsseltes HTTP (siehe [BelgRootCA]).

damit diese sie in der Zertifikatskette ausliefern können.

Ein Document Verifier hat ferner die Aufgabe, Terminal Certificate Requests entgegen zu nehmen und Terminal Certificates zu erstellen.

5.1.3 Abläufe einer Country Signing CA

Eine Country Signing CA wird wie folgt erzeugt:

- Generieren eines ECDSA-Schlüsselpaars unter Beachtung der Bedingungen aus 4.2.2.1 (Country Signing CA und Document Signer) ab Seite 44
- Erzeugen eines Zertifikats der CSCA
- Signieren des Zertifikats (Selbstsignatur)
- Verteilen des Zertifikats

Im Rahmen der Verteilung muss das CSCA-Zertifikat auf sicherem Weg in die Terminals gelangen. Die Terminals können das CSCA-Zertifikat nutzen, um die Echtheit von elektronischen Personalausweises zu überprüfen.

Darüber hinaus muss eine CSCA Document Signer zulassen und signieren.

5.1.4 Abläufe eines Document Signers

Ein Document Signer wird wie folgt erzeugt:

- Generieren eines ECDSA-Schlüsselpaars unter Beachtung der Bedingungen aus 4.2.2.1 (Country Signing CA und Document Signer) ab Seite 44
- Erzeugen eines Zertifikats des DS
- Signieren des Zertifikats (Signatur durch die CSCA)

Die Hauptaufgabe eines Document Signers besteht in der Erstellung, Personalisierung und Signatur von ePAs.

5.1.5 Erstellung und Personalisierung des ePA

Diese Ablaufbeschreibung bezieht sich auf die in Kapitel 4 (Komponentenbeschreibung) beschriebenen Komponenten.

Die Erstellung und Personalisierung des ePA gliedert sich in die folgenden fünf

Schritte:

1. Erfassung der personenbezogenen Daten
2. Erzeugen eines elektronischen Personalausweises
3. Personalisieren des ePA
4. Signieren des ePA
5. Serialisierung des ePA

Zu 1)
Zunächst müssen die personenbezogenen Daten zu einem Ausweis für eine bestimmte Person erfasst werden. Dies entspricht dem ersten Schritt. Hierbei wird im Rahmen der Beispiel-Implementierung die Menge der in [BSI-TR-03110] in Tabelle E.1 genannten Data Groups der personenbezogenen Daten erfasst (siehe Tabelle 7, S.67).

Zu 2)
Daraufhin muss eine Instanz eines elektronischen Personalausweises erstellt werden. Dies umfasst die Erzeugung des statischen ECDH-Schlüsselpaares **`keyPairPiccEcdhEpa`**. Die erzeugte Instanz des ePA trägt zu diesem Zeitpunkt noch keine personenbezogenen Daten.

Zu 3)
Die erzeugte Instanz des ePA wird mit personenbezogenen Daten personalisiert. Die Daten werden dabei in wieder beschreibbare Bereiche geschrieben, sodass sie im Nachhinein – vorausgesetzt ein Terminal besitzt die entsprechenden Rechte – geändert werden können.

Zu 4)
Der öffentliche Schlüssel **`pkPicc`** des statischen Schlüsselpaares **`keyPairPiccEcdhEpa`** wird durch den Document Signer signiert. Diese Signatur ist wichtig, damit Terminals beim Auslesen verifizieren können, dass es sich um einen authentischen ePA mit unverändertem öffentlichen Schlüssel handelt. Die Schritte 3) und 4) können auch in umgekehrter Reihenfolge ausgeführt werden. Es spielt für Schritt 4) keine Rolle, ob der ePA personalisiert ist, sondern notwendige Voraussetzung ist lediglich, dass das statische Schlüsselpaar bereits generiert wurde. In die Signatur fließen keine

personenbezogenen Daten ein. Ferner spielt es für die Personalisierung in Schritt 3) keine Rolle, ob der ePA einen signierten, öffentlichen Schlüssel (Resultat aus Schritt 4) hat.

Zu 5)
Der Inhalt des ePA wird serialisiert. Dieser Schritt bildet die Herausgabe des Ausweisdokuments ab. Im Rahmen der Beispiel-Implementierung kann nun die serialisierte Form genutzt werden, um einen speziellen ePA zu erzeugen.

Identifikator der Data Group	Inhalt
DG3	Document Number
DG4	Date of Issue
DG5	Date of Expiry
DG7	Given Names
DG8	Family Names
DG9	Maiden Name
DG10	Date of Birth
DG11	Place of Birth
DG12	Nationality
DG13	Sex
DG14	Eye Color
DG15	Height

Tabelle 7: Menge der zu erfassenden personenbezogenen Daten nach Table E.1 [BSI-TR-03110]

Dieser Ablauf wurde im Rahmen der Beispielimplementierung realisiert.

5.2 Die INIT-Phase

In der INIT-Phase signalisiert der Benutzer einer Webseite, dass er die eID Web-Authentisierung durchführen möchte. Dies kann beispielsweise durch einen Klick auf einen Link erfolgen. Auf der Clientseite wird daraufhin das Applet von der Serverseite geladen und initialisiert.
Das Applet initialisiert bei Bedarf das RF-Lesegerät und den ePA.
Darüber hinaus baut das Applet eine Verbindung zum eID Authentication Proxy auf.

Auf der Serverseite baut der EAP eine Verbindung zum EAS auf.
Die Komponenten sind nun bereit für den Ende-zu-Ende-Verbindungsaufbau zwischen ePA und EAS.

5.3 Die SETUP-Phase

Die SETUP-Phase ist die umfangreichste Phase. Ziel der SETUP-Phase ist der Aufbau eines sicheren Kanals zwischen dem elektronischen Personalausweis und dem eID Authentication Server. Hierzu werden in erster Linie die Protokolle PACE, Terminal Authentication und Chip Authentication zur Anwendung gebracht.
Das Sequenzdiagramm in Abbildung 22 (im Querformat auf Seite 72) verdeutlicht den Ablauf der SETUP-Phase.

Schritt A

Zunächst übermittelt der EAS sein Berechtigungs-Zertifikat (BZ) sowie die Zertifikatskette an den EAP.

Schritt B

Der EAP reicht beide an das Applet weiter. Im Applet kann wahlweise bereits eine zusätzliche, erste Verifikation der Zertifikate stattfinden.

Schritte C & D

Je nach Funktionalität des Lesegeräts führt entweder das Applet oder das Lesegerät das Protokoll Password Authenticated Connection Establishment (PACE) mit dem ePA durch. Nach dem erfolgreichen Ablauf des PACE wurde ein sicherer Kanal zwischen dem RF-Leser bzw. dem Applet und dem ePA aufgebaut.

Schritt E

Das Berechtigungszertifikat (Terminal Certificate) und die Zertifikatskette werden an den ePA gegeben.

Schritt F

Der ePA verifiziert die Zertifikatskette und überprüft so, ob es sich um ein authentisches Terminal Certificate handelt. Ein Beispiel für eine Verifikation der Zertifikatskette ist in Abschnitt 2.2.4.2 in [BSI-TR-03110] genannt. Diese Verifikation ist auch dann notwendig, wenn in Schritt 2 bereits eine Verifikation seitens des Applets stattfand.

Schritt G

Wenn die Verifikation erfolgreich verläuft, extrahiert der ePA den öffentlichen Schlüssel **PK_PCD** des EAS aus dem Terminal Certificate. Anschließend wird die Terminal Authentication eingeleitet.

Schritt H

Im Rahmen des PACE wurde ein flüchtiges ECDH-Schlüsselpaar erzeugt. Da der PACE derzeit nur in Form einer Simulation durchgeführt werden kann, wird im Rahmen der Beispiel-Implementierung das flüchtige Schlüsselpaar des ePA erst in diesem Schritt erzeugt. In einer Voll-Implementierung wird das Schlüsselpaar, das im Rahmen des PACE erzeugt wurde, weiterverwendet.

Schritt I

Die Zufallszahl **r_PICC**, die Domain Parameter **d_PICC** sowie der Hashwert des flüchtigen öffentlichen Schlüssels **~PK_PICC** des ePA werden an den EAS übertragen.

Schritt J

Der EAS generiert unter Verwendung der übertragenen Domain Parameter **d_PICC** ein kurzlebiges ECDH-Schlüsselpaar, bestehend aus dem geheimen Schlüssel **~SK_PCD_TA** sowie dem öffentlichen Schlüssel **~PK_PCD_TA**.

Schritt K

Der EAS signiert mit seinem statischen geheimen Schlüssel **SK_PCD** eine Nachricht, die aus den folgenden Bestandteilen zusammengesetzt wird:

- dem Hashwert des flüchtigen öffentlichen Schlüssels des ePA **~PK_PICC**
- der Zufallszahl **r_PICC** sowie
- dem Hashwert des flüchtigen öffentlichen Schlüssels des Terminals

~PK_PCD_TA

Schritt L

Der EAS übermittelt ein Tupel bestehend aus dem Hashwert des flüchtigen öffentlichen Schlüssels des Terminals **~PK_PCD_TA** sowie der in Schritt 9 erzeugten Signatur an den ePA.

Schritt M

Der ePA verifiziert die Signatur.

Schritt N

Der öffentliche statische Schlüssel des ePA **PK_PICC** wird in signierter Form an den EAS übertragen. **PK_PICC** muss dabei durch den Document Signer des ePA signiert worden sein. Dies setzt voraus, dass auf der Seite des Terminals der öffentliche Schlüssel des Document Signer **PK_DS** vorliegt. Sollte dies nicht der Fall sein, könnte der entsprechende öffentliche Schlüssel aus einem Zertifikat des Document Signer extrahiert werden. Dieses Zertifikat muss anhand des öffentlichen Schlüssels der CVCA **PK_CVCA** verifiziert werden. In jedem Fall muss der öffentliche Schlüssel der CVCA **PK_CVCA** im EAS vorliegen.
Es ist wichtig, dass dieser Schlüssel auf vertraulichem Weg in den EAS gelangt, z.B. durch eine Offline-Installation des CVCA-Zertifikats.

Schritt O

Der EAS verifiziert die Signatur des **PK_PICC** und führt einen Diffie-Hellman Schlüsselaustausch durch. In den Schlüsselaustausch fließen der flüchtige geheime Schlüssel **~SK_PCD_TA** des EAS sowie der öffentliche Schlüssel des ePA **PK_PICC** mit ein. Aus dem Ergebnis des Schlüsselaustauschs werden ein Verschlüsselungsschlüssel **K_ENC** sowie ein Integritätssicherungs-Schlüssel **K_MAC** abgeleitet.

Schritt P

Der EAS sendet seinen kurzlebigen öffentlichen Schlüssel **~PK_PCD_TA** an den ePA.

Schritt Q

Der ePA führt seinerseits den Diffie-Hellman-Schlüsselaustausch durch und erhält ebenfalls das gemeinsame Geheimnis. Aus dem gemeinsamen Geheimnis werden ein Verschlüsselungsschlüssel **K_ENC** sowie ein Integritätssicherungs-Schlüssel **K_MAC** abgeleitet. Darüber hinaus wird aus dem kurzlebigen öffentlichen Schlüssel **~PK_PCD_TA** des EAS mit dem Integritätssicherungs-Schlüssel anhand des Message Authentication Code ein Token **T_PICC** berechnet.

Schritt R

Das Token **T_PICC** wird vom ePA an den EAS gesendet.

Schritt S

Der ePA berechnet seinerseits das Token und vergleicht das Resultat. Erst wenn der Vergleich positiv ausfällt, ist die SETUP-Phase erfolgreich abgeschlossen.

Das Resultat der SETUP-Phase ist ein sicherer Kanal zwischen dem ePA und dem eID Authentication Server. In der folgenden Phase (READ-Phase) können personenbezogene Daten sicher ausgelesen werden.

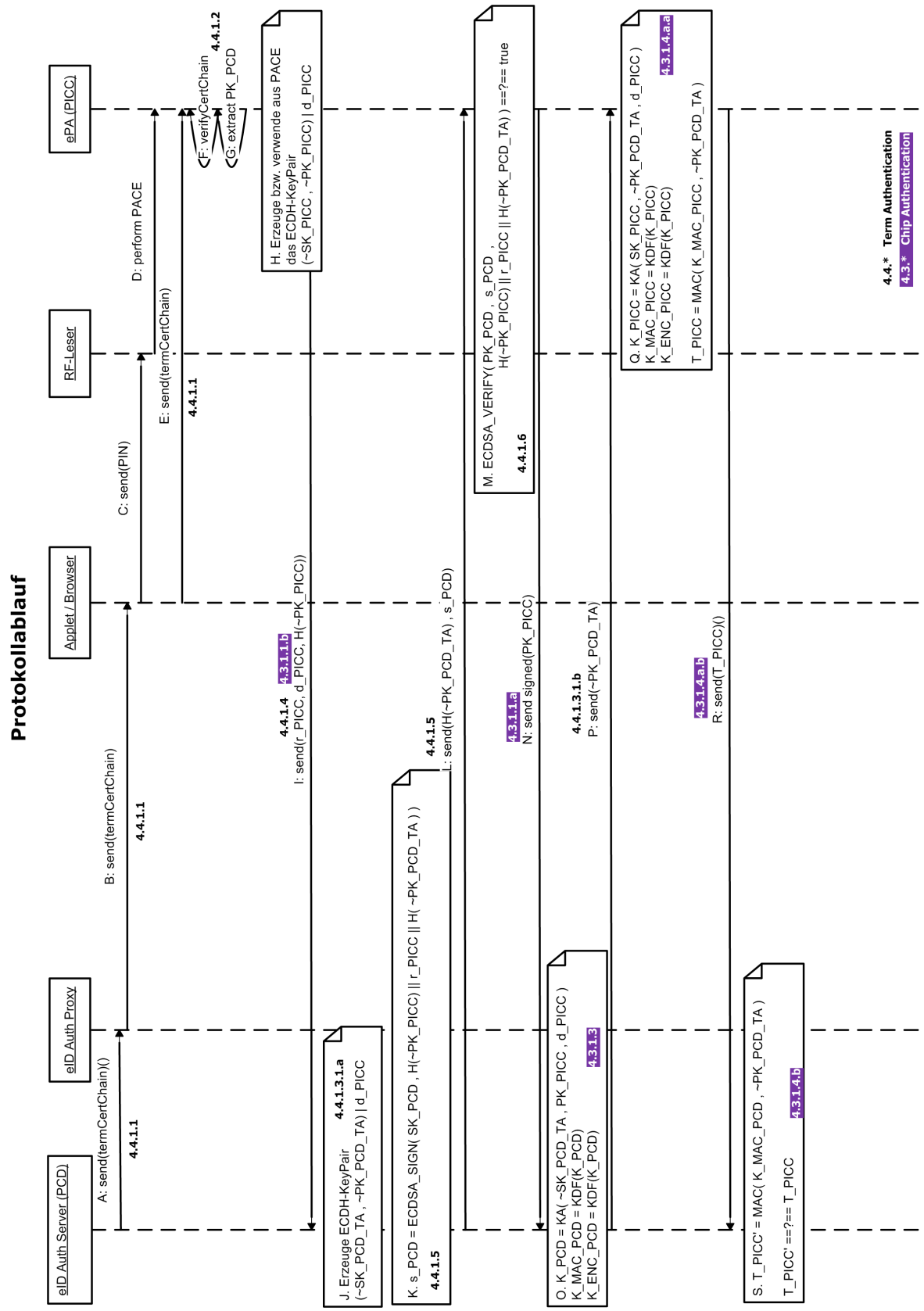


Abbildung 22: Protokollablauf von PACE, Terminal und Chip Authentication. Numerische Bezeichnungen stammen aus der Protokollspezifikation ([BSI-TR-03110]).

5.4 Die READ-Phase

Die READ-Phase ist die Phase, in der die personenbezogenen Daten ausgelesen werden. Diese Phase ist bereits durch den in der SETUP-Phase aufgebauten sicheren Kanal geschützt. Dritte können die ausgelesenen Daten weder mitlesen noch verändern.
Da es sich um einen verschlüsselten Kanal von EAS bis zum ePA handelt, müssen die beiden Endpunkte sozusagen den kleinsten gemeinsamen Nenner zur Kommunikation finden. Der EAS muss daher bereits verschlüsselte Command APDUs nach [ISO/IEC 7816-4] (vgl. [BSI-TR-03110]) versenden. In der READ-Phase handelt es sich dabei um die folgenden APDUs:

- SELECT FILE
- READ BINARY

Ein Lesevorgang in der READ-Phase, bei dem die Vor- und Nachnamen ausgelesen werden, könnte dann wie folgt aussehen:

```
SELECT FILE 0x0107
READ BINARY
SELECT FILE 0x0108
READ BINARY
```

Die Adresse 0x0107 bezieht sich auf die Data Group 7, die die Vornamen enthält. Die Adresse 0x0108 bezieht sich auf die Data Group 8, die die Nachnamen enthält. Die APDUs sind dabei im Rahmen des Secure Messaging verschlüsselt und signiert.

5.5 Die CLOSE-Phase

Die Verbindung zwischen dem ePA und dem EAS wird beendet, sobald das Auslesen der personenbezogenen Daten abgeschlossen ist. Mit der Übergabe der personenbezogenen Daten an den EAP wird dem Applet signalisiert, dass die Verbindung beendet werden soll.

Ferner wird die Verbindung abgebaut, sobald ein Fehler in der Authentisierung stattfindet. Der EAS informiert den EAP in jeder Response über den Zustand. Die Zustandsvariable hat einen der Werte IN_PROCESS, SUCCESS oder FAILURE. SUCCESS bedeutet, dass die personenbezogenen Daten erfolgreich ausgelesen werden konnten.
IN_PROCESS bedeutet, dass der Leseprozess noch nicht abgeschlossen ist (es werden weitere Schritte folgen).
FAILURE bedeutet, dass es einen Fehler beim Auslesen der des ePA gab. Die Verbindung muss unverzüglich abgebaut werden.

In der CLOSE-Phase teilt der EAS dem Applet eine URL mit, zu der das Applet einen Seitenwechsel veranlasst.

5.6 [gestrichen]

5.7 [gestrichen]

5.8 [gestrichen]

5.9 [gestrichen]

5.10 Schutz gegen High-Level-MITM

Ein zusätzlicher Schutzmechanismus zur Verhinderung von Man-in-the-middle-Attacken könnte die Bindung des Hostnamens des EAS an ein Terminal Certificate darstellen. Dieser Mechanismus wird bei den Implementierungen von Transport Layer Security (TLS) häufig angewendet und schützt auch dort vor Man-in-the-middle-Angriffen. Allerdings wird dieser Mechanismus nicht von TLS vorgeschrieben, sondern es handelt sich um ein Implementierungsdetail.
Im Fall von TLS vergleicht man den Common Name des X.509-Zertifikats des Servers mit dem Hostname, mit dem der Client verbunden ist. Weichen diese

beiden voneinander ab, besteht die Gefahr eines Man-in-the-middle-Angriffs.

> **Bemerkung**
> Der o.g. Schutzmechanismus im Rahmen von TLS gilt unter der Voraussetzung, dass auf den darunter liegenden Schichten kein Spoofing stattfindet (insbesondere beim Domain Name System oder beim Medienzugriff).

Für die Web-Authentisierung könnte man zusätzlich den Hostname des Terminals in das Terminal Certificate einbinden. Eine clientseitige Komponente kann nun – ähnlich wie bei TLS – den angegebenen Hostname des Terminal Certificate mit dem Hostname des Servers, mit dem die Verbindung zustande kommen soll, vergleichen. Bei einer Abweichung könnte dem Benutzer eine Warnung angezeigt werden oder die Verbindung zur Abwehr eines eventuellen Man-in-the-middle-Angriffs beendet werden.
Eine derartige Erweiterung des Terminal Certificate ist bisher nicht spezifiziert. Die Implementierung dieser Lösung könnte problematisch werden, da der ePA nicht direkt kontrollieren kann, mit welchem Host tatsächlich kommuniziert wird.[15] Die Kommunikation mit dem EAS bzw. dem EAP übernimmt auf der Clientseite das Applet. Im Fall eines bösartigen Applet könnte das Applet dem ePA gegenüber einen anderen Hostname präsentieren als tatsächlich in der Kommunikation verwendet wird.
Bei der Verwendung eines gutartigen Applets bietet diese Lösung jedoch einen zusätzlichen Schutz gegen einen Man-in-the-middle-Angriff gegenüber einem bösartigen Server. Ferner ist zu beachten, dass bei einer Implementierung unter Berücksichtigung der genannten Bedingungen ein Man-in-the-middle-Angriff selbst ohne diesen zusätzlichen Schutz scheitert.

15 Socket-Operationen sind Teil des Betriebssystems des Hosts und nicht des elektronischen Personalausweises.

5.11 Weitere Ansatzpunkte für Angriffe

Neben den in den vorherigen Abschnitten dokumentierten Angriffen sind die folgenden Ansatzpunkte für mögliche Angriffe herauszustellen:

- die Verifikation des Terminal Certificate durch den ePA sowie
- das Abwählen von Attributen

Die Verifikation des Terminal Certificate erfolgt im ePA. Der ePA hat eine innere, logische Uhr, die anhand von erfolgreichen Verifikationen in der Vergangenheit synchronisiert wird. Jede erfolgreiche Authentisierung synchronisiert automatisch die innere Uhr des ePA. Wenn die innere Uhrzeit vor dem Beginnzeitpunkt eines gültigen Zertifikats, das zur Authentisierung genutzt wird, liegt, wird die innere Uhrzeit auf den Beginnzeitpunkt der Gültigkeit des Zertifikats (Certificate Effective Date) gesetzt. Dies impliziert, dass alle Zertifikate nach dem selben Schema für den Gültigkeitszeitraum ausgestellt werden, z.B. mit einer Gültigkeit von etwa 24 Stunden, mit gleicher Start- und Endzeit.
Je seltener ein ePA benutzt wird (konkreter: je seltener erfolgreich authentisiert wird), umso größer ist die Abweichung der logischen Uhr des ePA von der Realzeit. Sobald das Schlüsselmaterial eines vormals legitimen EAS sowie ein in Bezug auf die Realzeit abgelaufenes Terminal Certificate für einen Angreifer verfügbar wird, könnte ein ePA mit hoher Abweichung seiner logischen Uhr von der Realzeit den Zugriff eventuell fälschlicherweise zulassen. Auch die Abfrage von Sperrlisten kann nicht verlässlich implementiert werden, da Socket-Operationen Teil des Betriebssystems des Clients und nicht des ePA sind. In der Konsequenz sollte der Gültigkeitszeitraum von Terminal Certificates sehr kurz ausfallen, z.B. lediglich 24 Stunden. Sobald Missbrauch festgestellt wird, kann auf diese Weise durch Verhinderung der Neuausstellung eines Zertifikats reagiert werden. Trotzdem gilt: Je häufiger ein ePA zur Authentisierung genutzt wird (und je geringer damit die Abweichung von Realzeit und logischer Uhr), umso geringer ist das Erfolgsrisiko für einen solchen Angriff.

Ein weiterer Ansatzpunkt ist die mangelnde Gewährleistung der Nicht-Übertragung von abgewählten Attributen. Die spezifizierten und verwendeten Protokolle stammen aus der Zeit, als nicht vorgesehen war, dass der Benutzer

auf die Menge an auszulesenden Attributen Einfluss hat. Vermutlich aufgrund datenschutzrechtlicher Anforderungen wurde die An- und Abwahl einzelner Attribute jedoch in das Konzept des ePA mit aufgenommen. Dies stellt jedoch in der Umsetzung möglicherweise eine Schwachstelle dar. Vorausgesetzt, das Terminal Certificate erlaube den Zugriff auf eine Menge M_A von Attributen. Per Default seien alle Attribute aus M_A zur Übertragung ausgewählt. Der Benutzer wähle nun ein Attribut ab, sodass sich eine echte Teilmenge M_A' ergibt. Wenn nun die Software der Benutzerschnittstelle kompromittiert ist (z.B. das Applet), könnte diese Teilmenge möglicherweise in veränderter Art, z.B. als die Obermenge M_A an den ePA weitergegeben werden. Die Software ignoriert also die Auswahl des Benutzers. Der ePA kann die ihm übergeben Menge an Attributen dann nur gegen die erlaubten Attribute des Terminal Certificate (also die Obermenge M_A) abgleichen. Auf diese Art könnte ein Attribut ausgelesen werden, dass der Benutzer abgewählt hatte. Voraussetzung hierbei ist jedoch, dass das Terminal Certificate dieses Attribute in den Berechtigungen auch tatsächlich enthält. Grundsätzlich sollten also nur die für die Transaktion absolut notwendigen Attribute im Terminal Certificate abgedeckt sein.

6 Beschreibung der Beispiel-Implementierung

Dieses Kapitel beschreibt wichtige Aspekte der Beispiel-Implementierung. Darüber hinaus bietet sie Hilfestellung für zukünftige, weitere Implementierungen von Extended Access Control.

6.1 Nachrichten-Sequenz

Die Nachrichten-Sequenz der Beispiel-Implementierung wird in Abbildung 23 auf Seite 79 dargestellt. Sie zeigt, dass das Applet das aktive Element ist. Das Applet stößt jeweils die Kommunikation an und verarbeitet die Rückgabewerte bzw. gibt sie weiter.

Für die Anwendung im Internet ist es erforderlich, dass das Applet eine aktive Rolle spielt, da sich der Client möglicherweise hinter einer Firewall befindet (siehe Abschnitt 4.5 ab Seite 49). Dies hat zur Konsequenz, dass das Applet eine Verbindung in Richtung EAP aufbauen kann, jedoch nicht umgekehrt. Ferner kommuniziert das Applet in der lokalen Umgebung mit dem ePA.

Bei der Nachrichten-Sequenz handelt es sich um eine Implementierung des Protokollablaufs der Phasen aus den Abschnitten 5.2 bis 5.5 auf den Seiten 67 bis 73.

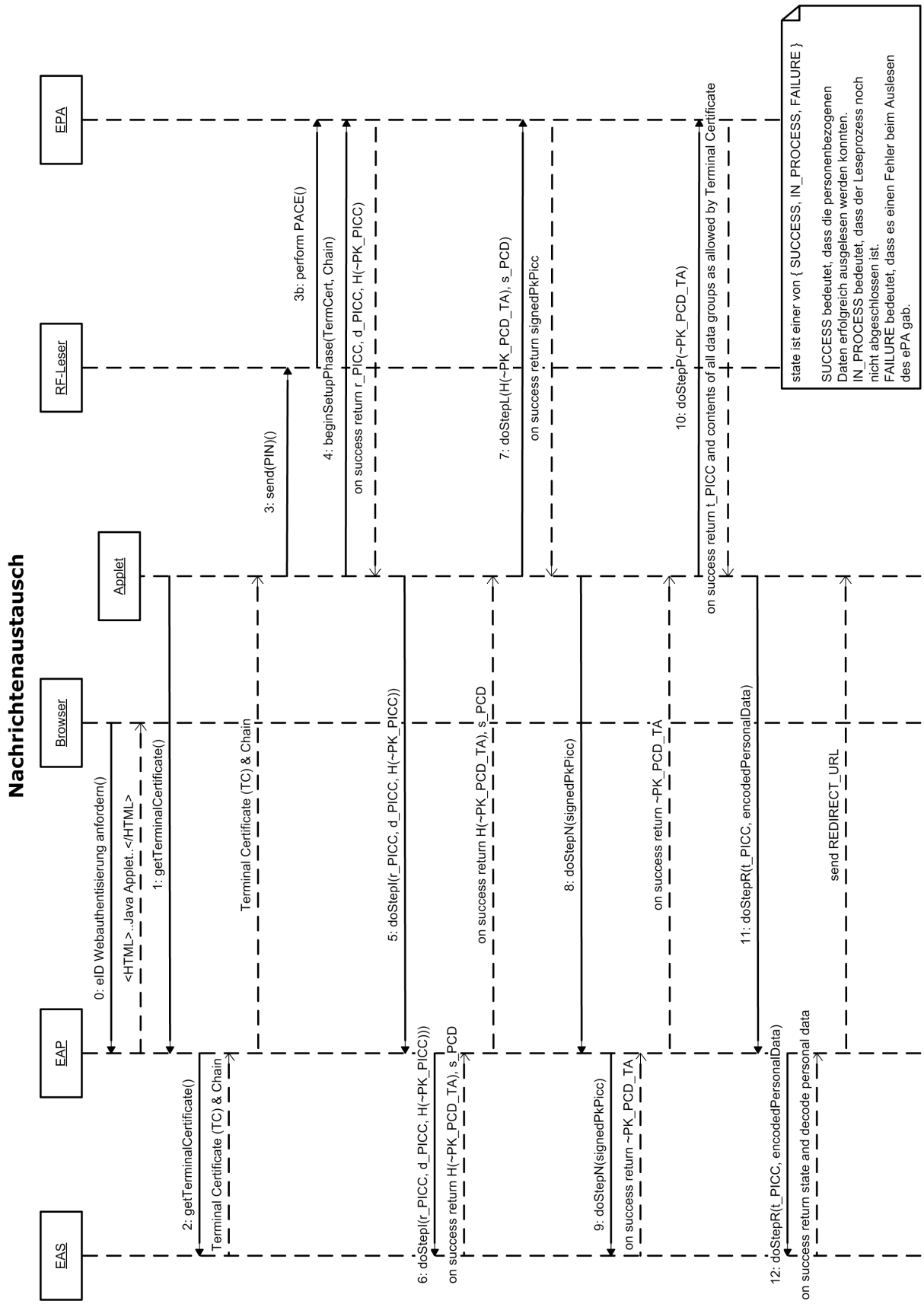


Abbildung 23: Nachrichten-Sequenz der Beispiel-Implementierung. Implementiert u.a. den Protokollablauf aus 5.3

Schritt 0

Der Benutzer (hier Browser) signalisiert dem Webserver, dass die eID Web-Authentisierung durchgeführt werden soll (INIT-Phase). Dies erfolgt beispielsweise durch den Klick auf einen entsprechenden Link. Der Server sendet daraufhin eine Seite, in die das Applet eingebunden ist. Das Applet beginnt automatisch mit dem nächsten Schritt (SETUP-Phase).

Schritt 1 & 2

Das Applet erfragt das Terminal Certificate beim EAP. Der EAP erfragt das Terminal Certificate wiederum beim EAS. Der EAS gibt das Terminal Certificate sowie die Zertifikatskette an den EAP zurück. Der EAP gibt diese an das Applet.

Schritt 3

Das Applet initiiert PACE zwischen dem Lesegerät und dem ePA.

Schritt 4

Das Applet übergibt das Terminal Certificate und die Zertifikatskette an den ePA, damit dieser sie verifizieren kann (Schritte F & G aus 5.3). Wenn dies erfolgreich verläuft, gibt der ePA ein Dreitupel bestehend aus **`r_PICC`**, **`d_PICC`** und dem Hashwert des flüchtigen, öffentlichen Schlüssels **`H(~PK_PICC)`** zurück.

Schritt 5

Das Dreitupel aus Schritt 4 wird an den EAP gesendet.

Schritt 6

Das Dreitupel wird vom EAP an den EAS weitergegeben. Der EAS entgegnet mit dem Signatur-Token der Terminal Authentication sowie dem Hashwert des flüchtigen, öffentlichen Schlüssels **`H(~PK_PCD_TA)`**. Das Signatur-Token wird in Schritt K aus 5.3 erzeugt.

Schritt 7

Das Applet gibt das Signatur-Token sowie den Hashwert des flüchtigen,

öffentlichen Schlüssels **H(~PK_PCD_TA)** an den ePA. Der ePA verifiziert die Signatur (Schritt M aus 5.3). Im Erfolgsfall gibt der ePA seinen durch den Document Signer signierten statischen öffentlichen Schlüssel **signedPkPicc** zurück an das Applet.

Schritt 8

Das Applet sendet den signierten, statischen öffentlichen Schlüssel **signedPkPicc** an den EAP.

Schritt 9

Der EAP gibt **signedPkPicc** weiter an den EAS: Daraufhin überprüft der EAS die Signatur von **PK_PICC** und gibt seinen flüchtigen, öffentlichen Schlüssel aus der Terminal Authentication **~PK_PCD_TA** zurück. Dieser wird vom EAP an das Applet zurückgegeben.

Schritt 10

Das Applet gibt **~PK_PCD_TA** an den ePA weiter. Der ePA führt dann Schritt Q aus 5.3 aus. Im Erfolgsfall wird **t_PICC** zurück an das Applet gegeben. Darüber hinaus kann der ePA bereit hier die personenbezogenen Daten in verschlüsselter Form übertragen (Secure Messaging). Das Terminal kann die verschlüsselten Informationen nur dann korrekt entschlüsseln, wenn die Setup-Phase abgeschlossen ist (Schritt 12)

Schritt 11

Das Token **t_PICC** sowie die verschlüsselten personenbezogenen Daten werden vom Applet an den EAP weitergegeben.

Schritt 12

Das EAP leitet das Token und die verschlüsselten personenbezogenen Daten an den EAS. Im EAS wird das Token überprüft (Schritt S aus 5.3). Falls die Überprüfung erfolgreich ist, wird der sichere Kanal von Ende zu Ende (EAS zu ePA) aufgebaut. Alle weiteren Nachrichten werden verschlüsselt übertragen. Der EAS liest nun die personenbezogenen Daten aus dem ePA.

Der EAP gibt die Nachricht weiter an den EAS. Der EAS kann die

verschlüsselte Nachricht entschlüsseln und extrahiert die personenbezogenen Daten. Er signalisiert dem EAP, dass die READ-Phase erfolgreich war und gibt die personenbezogenen Daten an den EAP. Der EAP kann die personenbezogenen Daten nun in der Sitzung mit dem Benutzer weiterverwenden, zum Beispiel zum Abschluss einer Bestellung. Anschließend sendet der EAP dem Applet eine URL, auf die das Applet einen Seitenwechsel unternimmt. Im Fehlerfall könnte diese Seite detaillierte Informationen zum Fehler enthalten. Im Erfolgsfall zeigt die Seite beispielsweise eine Bestätigung der Transaktion.
Mit der Zustellung dieser letzten Nachricht (Seitenwechsel) in der Response von Schritt 12 wird die Verbindung automatisch abgebaut (CLOSE-Phase), es finden keine weiteren Zugriffe auf den ePA innerhalb dieser Sitzung statt.

6.2 Pakete und Komponenten

Die folgenden Abschnitte beschreiben Pakete und Komponenten der Beispiel-Implementierung. Die Paketstruktur der Beispiel-Implementierung ist in Abbildung 24 dargestellt.

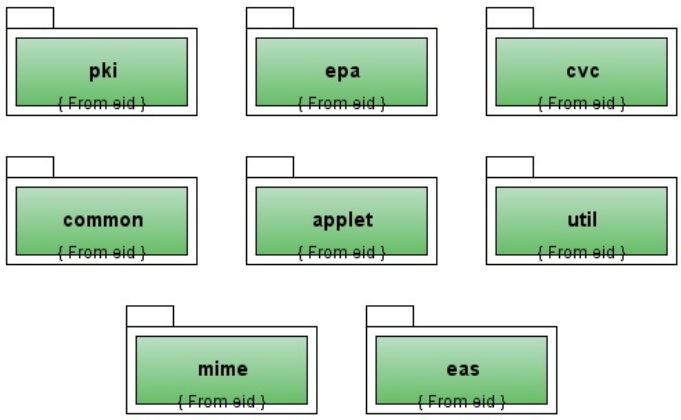


Abbildung 24: Pakete innerhalb des Wurzelpakets de.ifis.eid

Das Wurzelpaket de.ifis.eid besteht aus 8 Paketen. Ausschlaggebend für die Aufteilung in die Subpakete ist in erster Linie die Gruppierung nach Komponenten. Die von der Implementierung her umfangreichsten Komponenten sind der ePA, der EAS sowie das Applet.

Zunächst wurde ein gemeinsames Basispaket entwickelt, das grundlegende Funktionen für diverse Komponenten enthält. Dieses Paket wird als common bezeichnet. Es beinhaltet unter anderem die abstrakte Klasse EcdsaKeyStore, die als Basis für die Implementierung eines Document Signers sowie weiteren Klassen im Paket de.ifis.eid.epa dient. Sie bietet in erster Linie Funktionalität zum Serialisieren von Elliptic-Curve-DSA-Schlüsseln sowie zur Preisgabe eines CV-Zertifikats.

Darüber hinaus enthält das Paket häufig verwendete Exceptions, Steuerungsklassen für den Ablauf im Rahmen der Simulation sowie Klassen zur Kapselung von Domain Parametern für ECC.

Alle Klassen mit Bezeichnern nach dem Muster **`Main*`** sind Steuerungsklassen und führen die Simulation auf verschiedene Arten aus. **`MainObjects`** sorgt dafür, dass lediglich die Schicht „Objekte“ des Schichtenmodells aus Abschnitt 6.6 benutzt wird. Die Simulation ist dann nur auf einem lokalen Host, innerhalb derselben Java Virtual Machine lauffähig. Es findet keine Netzwerk-Kommunikation statt. Die Klasse **`MainByte`** limitiert die Ausführung auf die beiden Schichten „Objekte“ und „Byte-Arrays“. Die Klasse **`MainAppletSimulation`** sorgt in der Ausführung dafür, dass alle Schichten verwendet werden. Auf die Details des Schichtenmodells wird in Abschnitt 6.6 eingegangen.

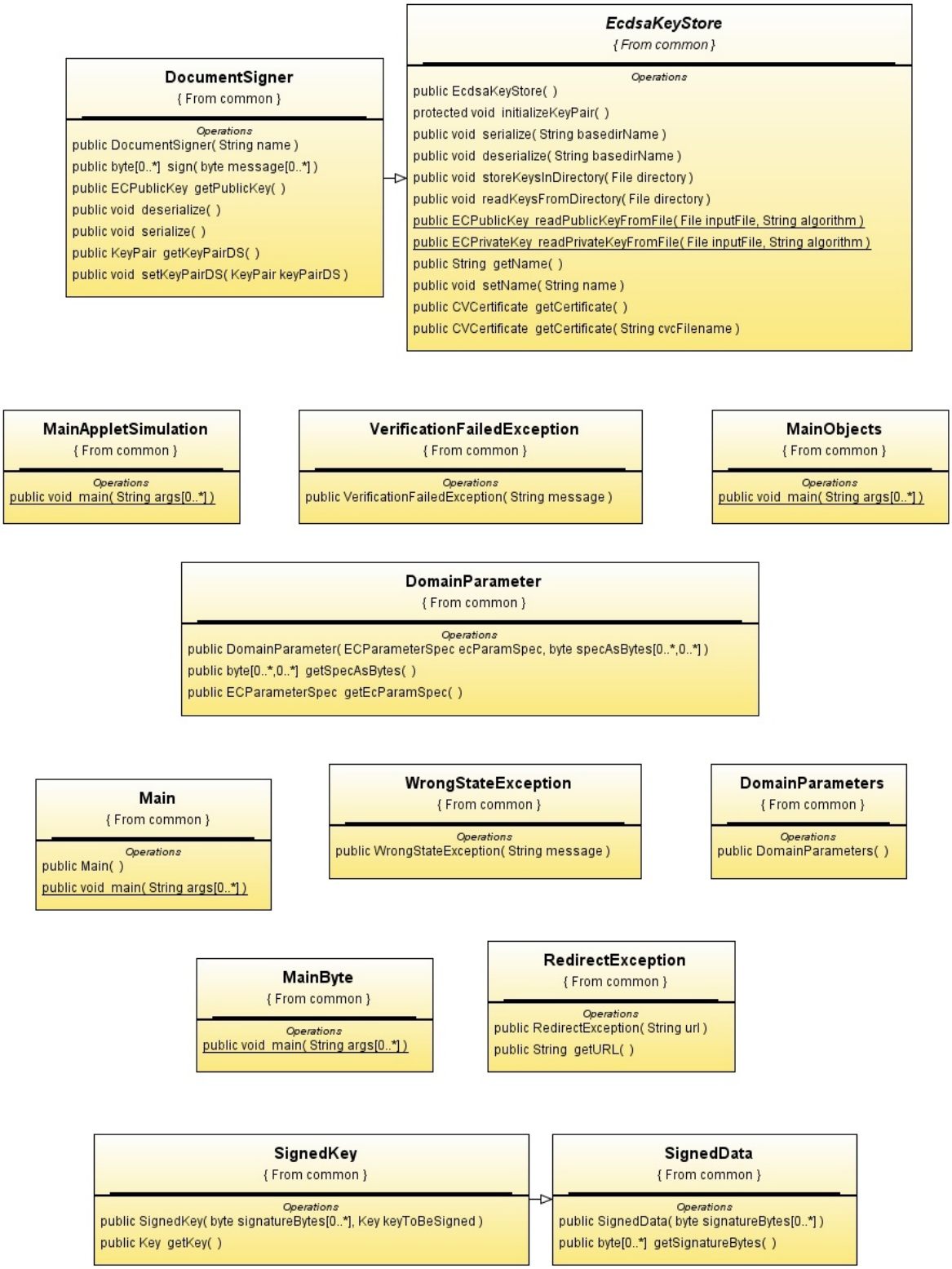


Abbildung 25: Klassendiagramm des Pakets de.ifis.eid.common. Aus Platzgründen sind hier nur die wichtigsten Klassen dargestellt und infolgedessen auch nicht alle Relationen sichtbar.

6.3 Public Key Infrastructure

Die Beispiel-PKI wurde in der Programmiersprache Java implementiert. Hierzu wurden die folgenden Klassen entworfen und umgesetzt:

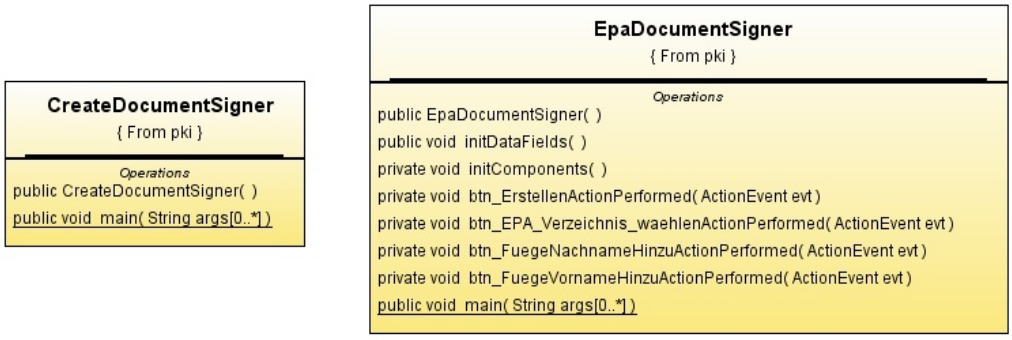


Abbildung 26: Klassendiagramm des Pakets de.ifis.eid.pki. Aus Platzgründen sind hier nur beispielhaft 2 herausgenommen.

Die Klasse EpaDocumentSigner beinhaltet eine grafische Oberfläche eines DocumentSigners, mit Hilfe derer ein elektronischer Personalausweis ausgestellt werden kann (siehe Abbildung 27).

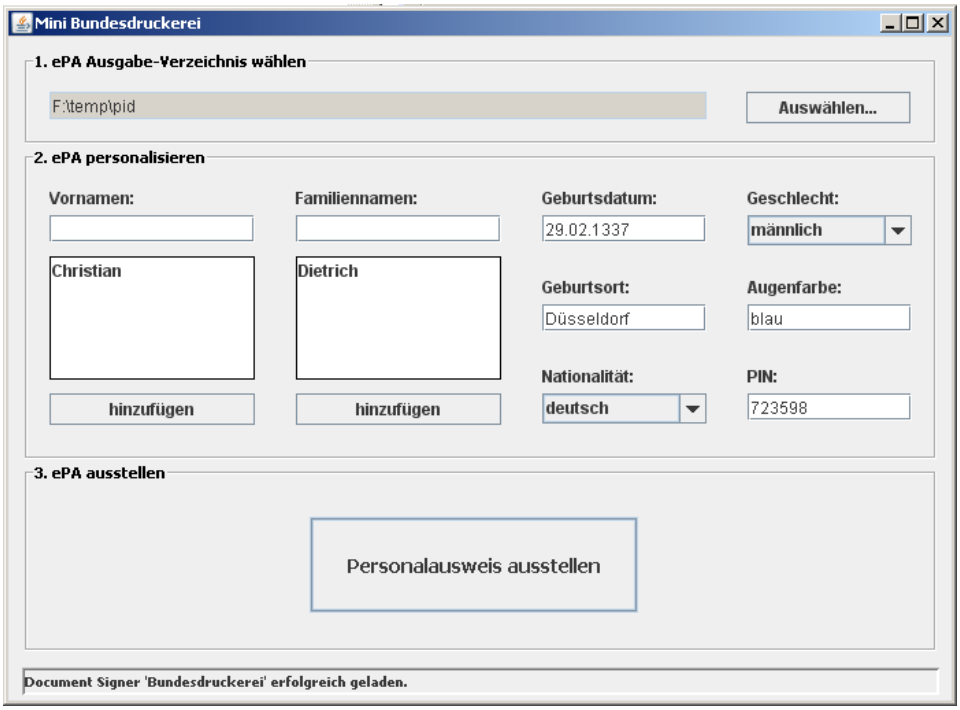


Abbildung 27: Screenshot der grafischen Oberfläche des Document Signer der Beispiel-Implementierung

Für die Funktion, Card Verifiable Certificates zu signieren wurde ein Teil des FlexSecure CV Reference Tools (siehe [FlexSecure CVREF]) wiederverwendet. Da ausschließlich ein geringer Teil der Funktionalität des CV Reference Tools frei verfügbar ist und die Datenhaltungsklassen Teil des kommerziellen Produkts FlexiTrust sind, wurden eigene Datenhaltungsklassen für Card Verifiable Certificates entworfen und implementiert. Diese befinden sich in dem Paket de.ifis.eid.cvc (siehe Abbildung 28).

Im Kern des Pakets liegt die Klasse **`de.ifis.eid.cvc.TLVData`**. Sie kapselt beliebige Daten im Tag-Length-Value-Format (TLV) und stellt grundlegende Parsing-Funktionalität für Tag-Length-Value-Daten zur Verfügung. Ferner bildet sie die Möglichkeit ein CVC-Zertifikat zu synthetisieren.
Alle erbenden Klassen sind Konkretisierungen für Objekte innerhalb eines Card Verifiable Certificate (CVC). Viele bieten die Möglichkeit an, die Nutzdaten als Objekte zurückzuliefern. Beispielhaft soll hier die Klasse **`de.ifis.eid.cvc.PublicKey`** herausgegriffen werden. Sie liest die diversen Parameter eines serialisierten, öffentlichen Schlüssels als TLV-formatierte Daten aus dem CVC und wandelt ihn in eine Instanz einer Klasse aus der Java Cryptography API, sodass das Resultat für Kryptographie-Operationen benutzt werden kann. In dieser Implementierung wird das Resultat als ein Objekt zurückgegeben, dass das Interface **`java.security.interfaces.ECPublicKey`** implementiert und damit zur Verwendung mit einer Java Cryptography Extension (JCE) gedacht ist.

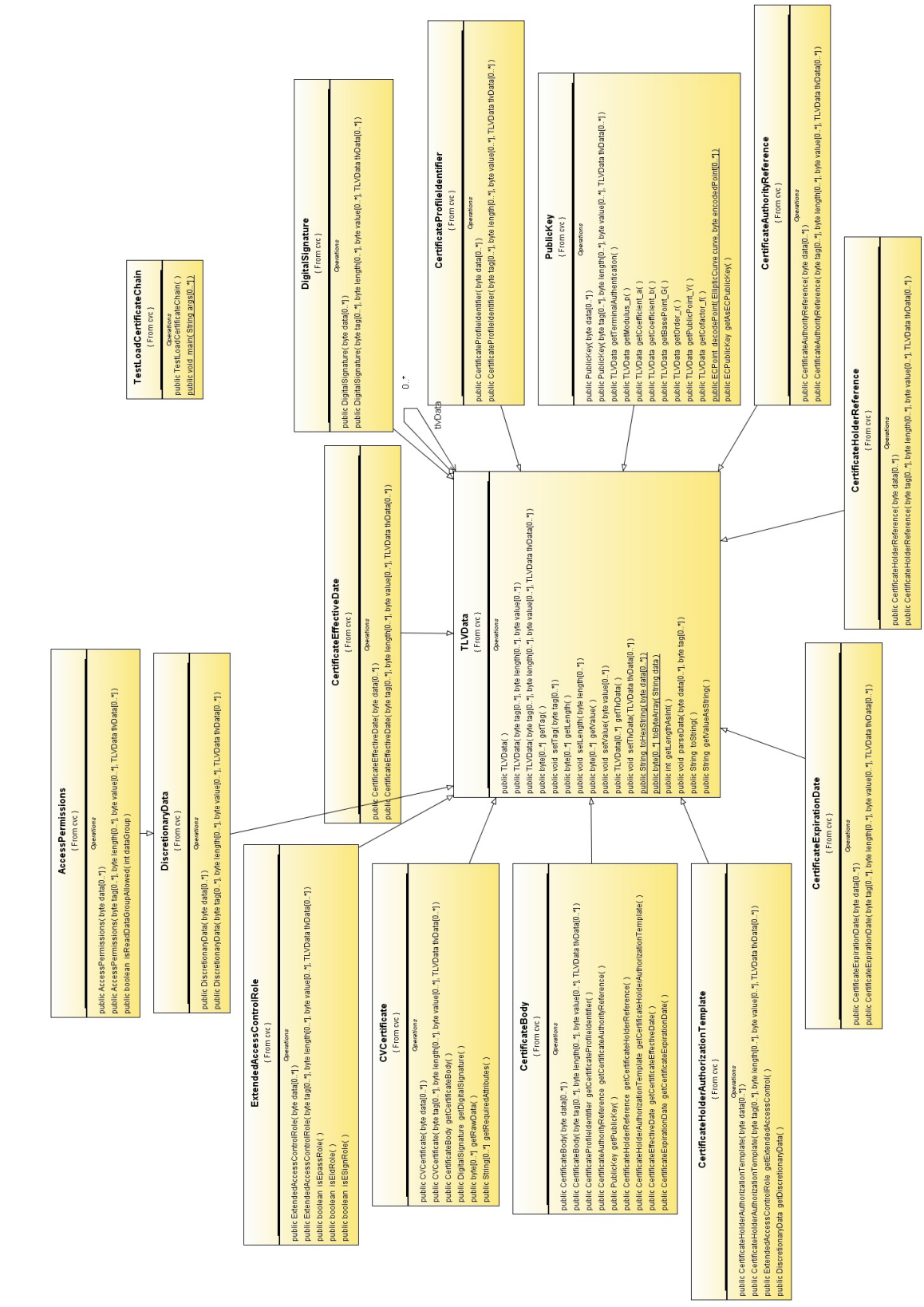


Abbildung 28: Klassendiagramm des Pakets de.ifis.eid.cvc

6.4 ePA/RF-Leser-Simulation

Der ePA und der RF-Leser wurden als eine Komponente realisiert. Aufgrund der Tatsache, dass es sich lediglich um eine Simulation handelt, muss die Schnittstelle zu ISO 14443 nicht implementiert werden. Die ePA/RF-Leser-Simulation wurde in der Programmiersprache Java realisiert. Kryptographische Operationen wurden mit Hilfe der Java Kryptobibliothek bouncycastle (siehe [Boundycastle Java API]) umgesetzt. Bouncycastle bietet – als zum derzeitigen Stand eine der wenigen – Unterstützung für Elliptic Curve Cryptography. Folgende Funktionalität wurde aus der Bibliothek bouncycastle verwendet:

- Schlüsselaustausch per ECDH
- Signieren und Verifizieren mittels ECDSA
- symmetrische Ver- und Entschlüsselung per AES
- Berechnung und Überprüfung des Message Authentication Codes CMAC-AES

Im Rahmen der ePA/RF-Leser-Simulation wurden die in Abbildung 29 dargestellten Klassen entworfen und umgesetzt.

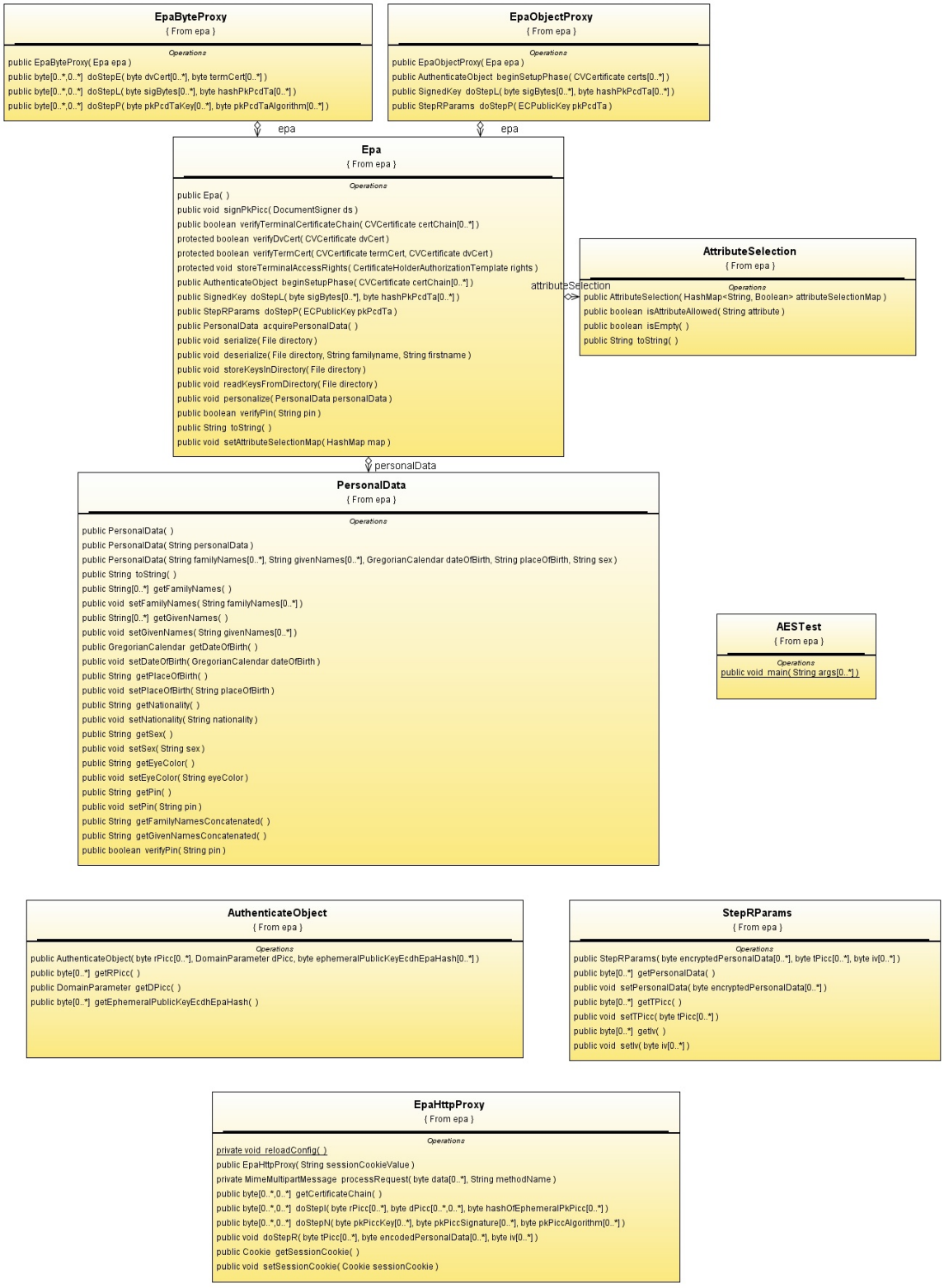


Abbildung 29: Klassendiagramm des Pakets de.ifis.eid.epa

6.5 eID Authentication Server (EAS)

Der eID Authentication Server wurde ebenfalls in der Programmiersprache Java implementiert. Er verwendet – wie die Komponente ePA/RF-Leser-Simulation – die Kryptobibliothek bouncycastle für kryptographische Operationen.

Im Rahmen des eID Authentication Servers wurden die folgenden Klassen entworfen und umgesetzt:

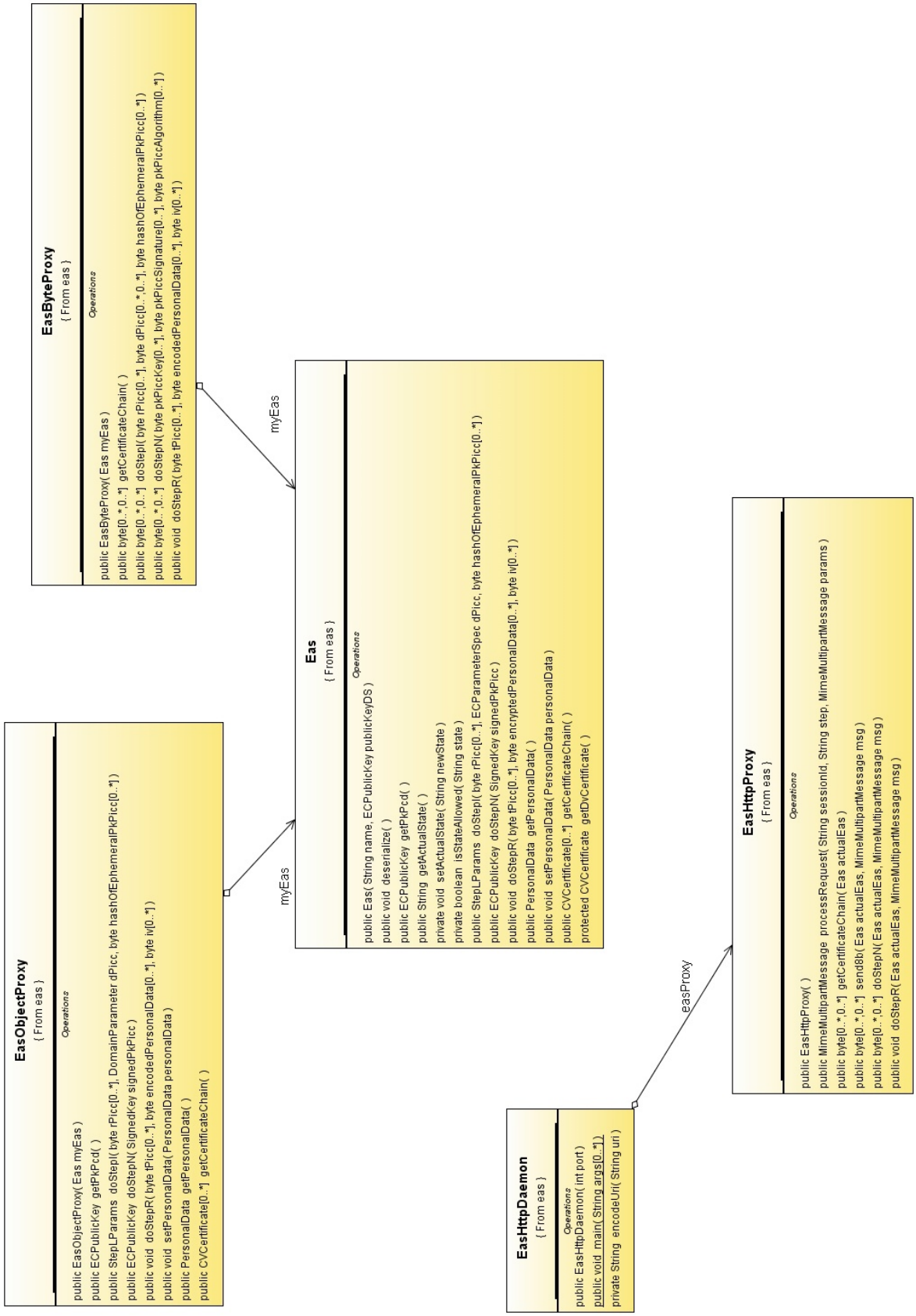


Abbildung 30: Klassendiagramm des Pakets de.ifis.eid.eas

6.6 Schichtenmodell

Für die Kommunikation zwischen dem ePA und dem EAS wurde ein Schichtenmodell entworfen. Die Daten haben auf den verschiedenen Schichten unterschiedliche Repräsentationen.

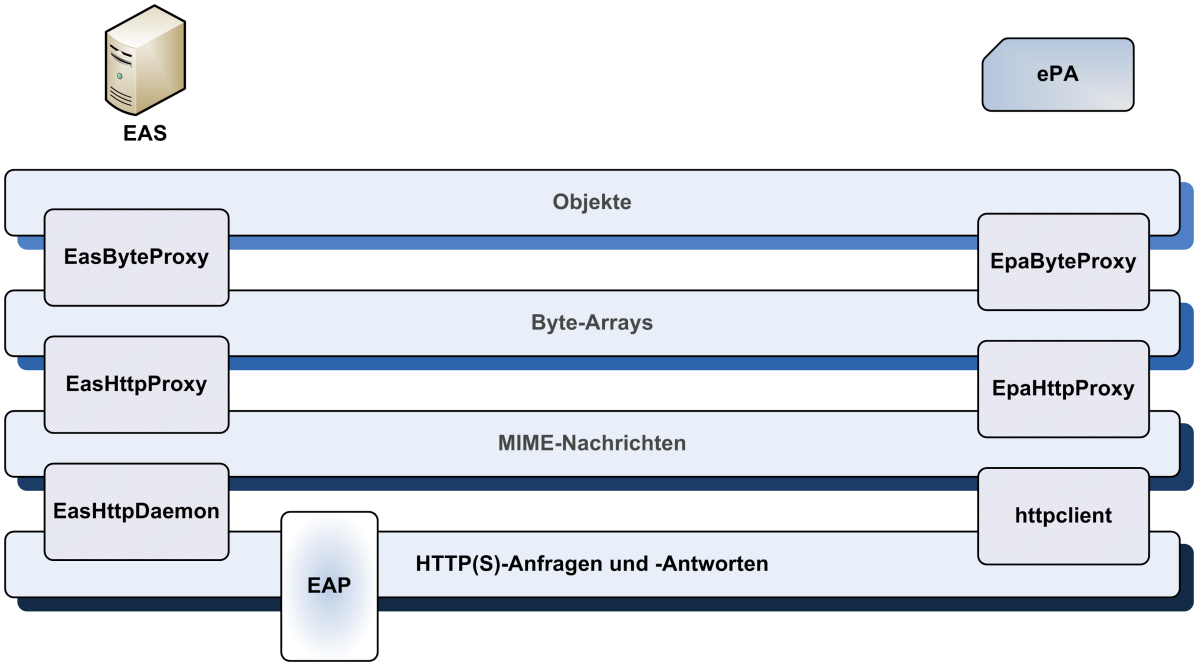


Abbildung 31: Schichtenmodell für die Kommunikation zwischen EAS und ePA

Auf der obersten Schicht wird objekt-orientiert gearbeitet. Sowohl die Komponenten als auch die Daten sind als Objekte realisiert. Die Kommunikation erfolgt durch Methodenaufrufe. Auf der zweiten Schicht werden die Daten als Byte-Arrays aufgefasst. Hierzu konvertieren auf beiden Seiten Adapter-Klassen (**`EpaByteProxy`** und **`EasByteProxy`**) die Daten in die entsprechende Form. Um die Daten über das Netzwerk transportieren zu können, werden die Byte-Arrays in MIME-Parts gekapselt (siehe Abschnitt 4.6.1.1). Diese Aufgabe übernehmen wieder spezielle Klassen, nämlich **`EpaHttpProxy`** und **`EasHttpProxy`**. Die MIME-Parts werden zu einer MIME-Nachricht zusammengefasst, die in Form eines HTTP-Requests oder einer HTTP-Response übermittelt wird. Auf der Seite des EAS wird dies durch die Klasse **`EasHttpDaemon`** erledigt, die als HTTP-Serverprozess implementiert ist. Auf der Seite des ePA, der auf HTTP-Ebene als Client agiert, wird die

HTTP-Client-Bibliothek **`httpclient`** des Apache-Frameworks (vgl. [Apache httpclient]) benutzt.

Das Schichtenmodell bietet den Vorteil, dass in sich abgeschlossene Aufgaben definiert und unabhängig voneinander implementiert werden können. Die Klassen behalten dadurch einen geringen Umfang und bleiben überschaubar. Insbesondere können zu Testzwecken der Kernkomponenten EAS und ePA die unteren Schichten vorübergehend umgangen werden und der Ablauf kann direkt auf oberster Ebene, also objekt-orientiert, ausgeführt und getestet werden.
Darüber hinaus muss der EAP lediglich Daten der untersten Schicht, in diesem Fall HTTP, interpretieren und verarbeiten können. Auch an dieser Stelle wird Komplexität vermieden.
Die Umsetzungsklassen zwischen den Schichten sind nach dem Entwurfsmuster **`Adapter`** entworfen (vgl. [GoF 1995]). Für den Fall, dass ein anderes Netzwerkprotokoll anstelle von HTTP verwendet werden soll, muss nur die Adapter-Klasse zur untersten Schicht angepasst werden. Wenn darüber hinaus das MIME-Format für Nachrichten nicht beibehalten werden soll, reicht eine Anpassung der Adapter-Klassen der untersten beiden Schichten. Die Schnittstellen der Umsetzungsklassen von einer Schicht zur nächsten sind in Schnittstellenbeschreibungen (in Java: Interface) ausgelagert.

6.7 Applet

Das Applet wurde in der Programmiersprache Java implementiert. Java-Applets sind eine weit-verbreitete Methode, um Code auf der Clientseite in einer sicheren Umgebung auszuführen. Hierzu bietet Java eine Menge an Sicherheitsfunktionen. Ein Java-Applet wird grundsätzlich in einer Sandbox, der sog. Java Virtual Machine ausgeführt (vgl. [JavaSecArch]). Ein unsigniertes Java-Applet hat zunächst ausschließlich folgende Berechtigungen:

- Netzwerkverbindungen dürfen nur zu dem Host aufgebaut werden, von dem das Applet geladen wurde.
- Lokale Dateien auf dem ausführenden Client dürfen weder gelesen noch geschrieben werden.

- Es dürfen keine Bibliotheken geladen oder native Funktionen definiert werden.
- Es dürfen keine Programme auf dem ausführenden Client gestartet werden.
- Es dürfen keine System-Eigenschaften ausgelesen werden.

Ein unsigniertes Applet hat daher keine Möglichkeit einen ePA über einen RF-Leser anzusteuern. Als Konsequenz ergibt sich die Notwendigkeit, ein Java Applet zu signieren, um es im Rahmen der Web-Authentisierung zu verwenden. Ein signiertes Applet hat weitere Vorteile, die in Abschnitt 4.5.5 (Vertrauenswürdigkeit des Applets) ab Seite 53 erläutert werden.

Für die Beispiel-Implementierung wird die Signatur des Applets wie folgt durchgeführt:

1. Mit Hilfe des keytool (siehe [JavaKeytool]) wird ein Schlüsselpaar erstellt.
2. Mit Hilfe des jarsigner (siehe [JavaJarsigner]) wird das Applet signiert.
3. Das keytool wird genutzt, um ein selbst-signiertes Zertifikat zu erstellen.
4. Damit der Browser das erzeugte Zertifikat als vertrauenswürdig anerkennt, muss es als vertrauenswürdiges Zertifikat in den Keystore des Java-Plugins importiert werden.

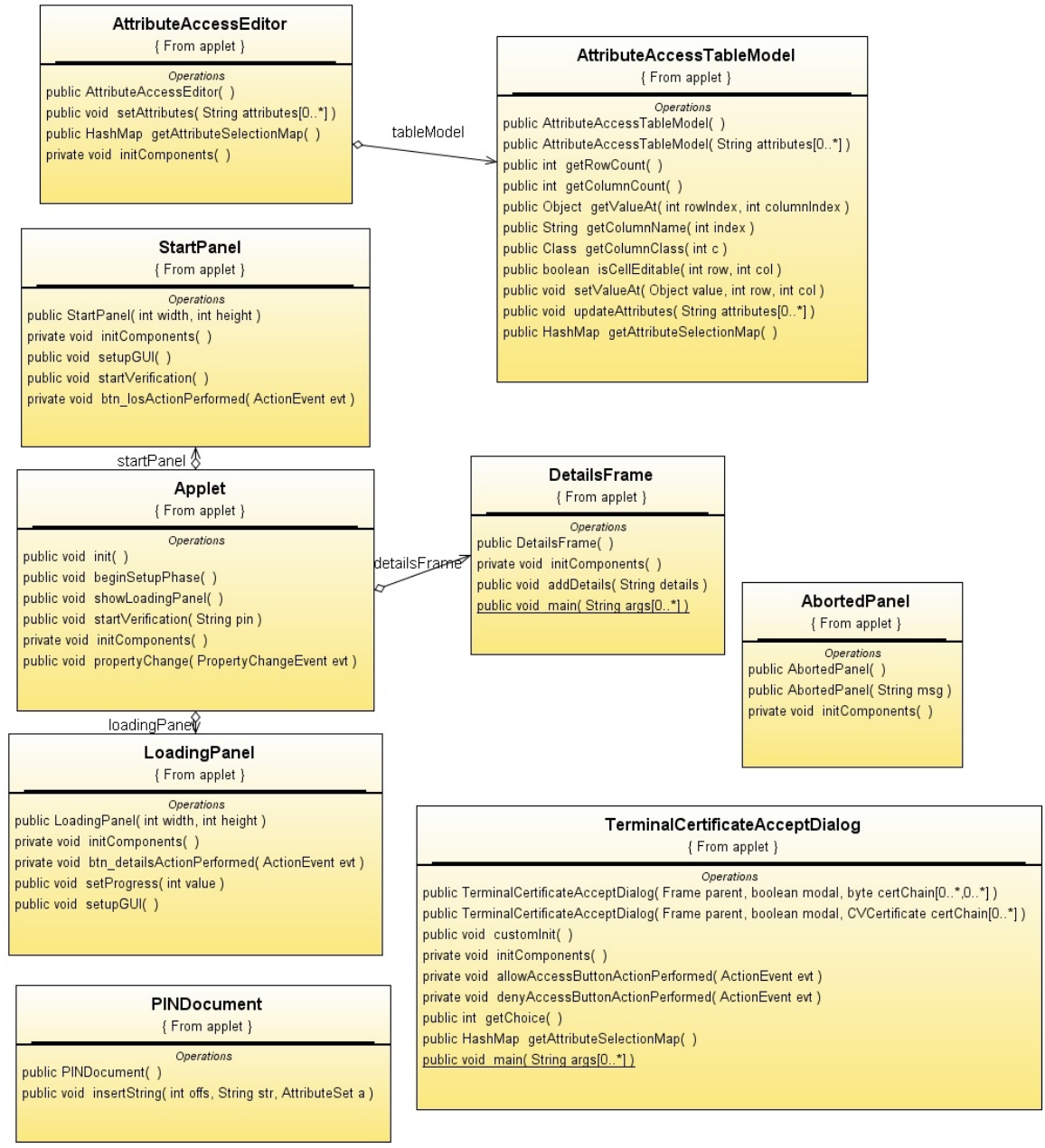


Abbildung 32: Klassendiagramm des Pakets de.ifis.eid.applet

6.8 eID Authentication Proxy (EAP)

Der eID Authentication Proxy ist eine Vermittlungskomponente auf der Serverseite. Sie schlägt die Brücke zwischen der Authentisierung durch den EAS und der Sitzung zwischen Webserver und dem Benutzer.

Für die Beispiel-Implementierung wurde der EAP als Erweiterung des freien Content-Management-Systems Typo3 ([Typo3]) implementiert. Dies wurde in der Programmiersprache PHP ([PHP]) realisiert. Bei der Verarbeitung der HTTP-Pakete wurde die HTTP-Bibliothek pecl_http ([PECL_HTTP]) eingesetzt. Der EAP nimmt dabei die Nachrichten vom Applet entgegen und gibt sie an den EAS weiter. Die Nachrichten bestehen aus HTTP-Paketen mit MIME-Bodies. Die Nutzdaten (MIME) werden auf dem Weg zum EAS nicht verändert. Der EAS liefert ebenfalls eine MIME-kodierte HTTP-Nachricht als Antwort. Diese Antwort enthält unter anderem den aktuellen Zustand des EAS, den der EAP ausliest. Anhand des Zustands erkennt der EAP, ob die Verbindung zum Applet weiterhin bestehen soll, abgebaut oder reinitialisiert werden soll.

Im Erfolgsfall kann dem EAP Zugriff auf die personenbezogenen Daten gewährt werden, damit dieser sie beispielsweise an das Content Management System weiterreichen kann. Im Rahmen eines Online-Einkaufs könnte ein Webshop die Daten erhalten, um die Bestellung abzuschließen.

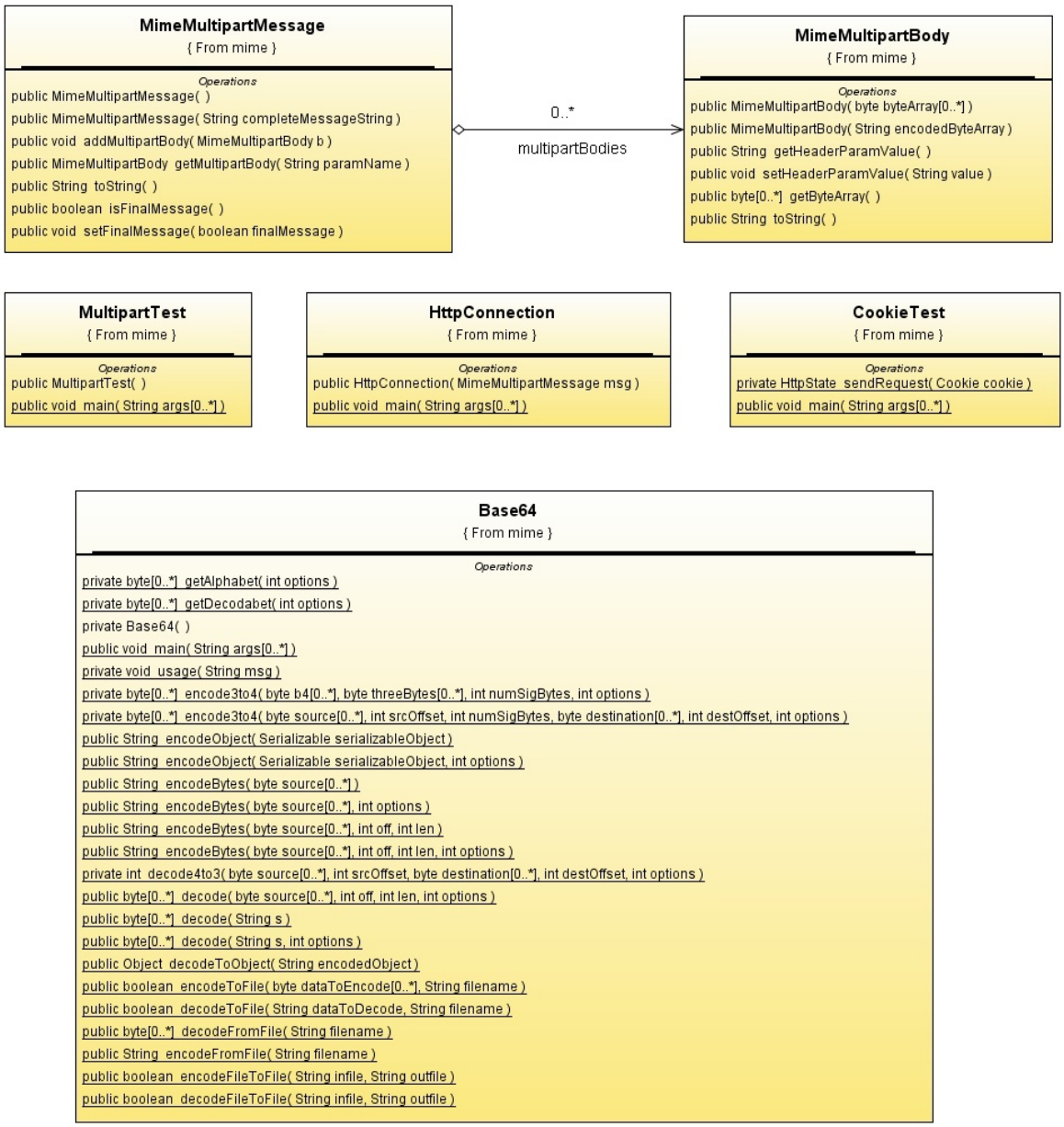


Abbildung 33: Klassendiagramm des Pakets de.ifis.eid.mime

6.9 Ablauf der Authentisierung auf HTTP-Ebene

Die folgenden Abschnitte zeigen die 4 Request-Response-Zyklen einer erfolgreichen Web-Authentisierung des Beispiel-Szenarios. Der Mitschnitt entstand auf der Strecke zwischen dem Applet und dem EAP. Orange hinterlegte Absätze entsprechen HTTP-Anfragen, die das Applet an den EAP schickt. Türkis hinterlegte Absätze entsprechen den HTTP-Antworten, die der EAP an das Applet zurückgesendet hat. Die Prozessschritte sind anhand des HTTP-Headers **`X-Eid-Authenticationmethod`** identifizierbar, der entsprechende Wert entspricht der Bezeichnung aus Abbildung 23. Die MIME-Part-Trenner wurden zur besseren Lesbarkeit alle auf **`--trenner`** gesetzt.

6.9.1 Erster Request-Response-Zyklus

```
POST /typo3conf/ext/is_eap/eapGateway.php HTTP/1.1
X-Eid-Authenticationmethod: getTerminalCertificate
User-Agent: Jakarta Commons-HttpClient/3.1
Host: www.internet-sicherheit.de
Cookie: $Version=0; PHPSESSID=enmkitol5j411fescvq4phjpr1
Content-Length: 0
```

```
HTTP/1.1 200 OK
Date: Sun, 06 Jul 2008 18:17:02 GMT
Server: Apache
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Content-Length: 1367
Connection: close
Content-Type: text/html; charset=UTF-8

MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="trenner"

--trenner
Content-Type: application/octet-stream;
.name="dvCert"
Content-Transfer-Encoding: base64

fyGCAYp/ToIBSl8pAQBCDkRFX0NWQ0FfQlNJXzAxf0mB/QYKBAB/AAcCAgICAoEc18E0qiZDZoYq
GDAlddHXh7CfB1eX2on1fsjA/4IcaKXmLKnObBwpmAOmwVMLUU4YKtiwBCpZytKfQ4McJYD2PM/k
QTiHBxOxqSNp4z4hNdJm27NyOGxAC4Q5BA2QKa0sflz0NAgjsqh9xoyeTOMXTB5u/e4SwH1Yqlb3
csBybyTGuJ5OzawkNUuemcqj9tN2FALNhRzXwTSqJkNmhioYMCV10PuY0Ra8S23evKOlp5OfhjkE
IH80iTAByxY5/kcsI/A9a2HKsPcnb88UZ17yHGRlzfF/v+kNnitEGmk9DzW90INWn5BXoeO9YvCH
AQFfIA9ERV9EVl9EVHJ1c3RfMDF/TA4GCQQAfwAHAwECAlMBgV8lBgAHAQEAAV8kBgAHAQIDAV83
ODOSA11rNEETWGwtsW5ejII8MnWPL93arrc/mqld4ezDuatR7Q9Pvou4PJxl43rV6b1VckPxT5RA

--trenner
Content-Type: application/octet-stream;
.name="termCert"
Content-Transfer-Encoding: base64

fyGCAYh/ToIBSF8pAQBCD0RFX0RWX0RUcnVzdF8wMX9Jgf0GCgQAfwAHAgICAgKBHNfBNKomQ2aG
KhgwJXXR14ewnwdXl9qJ9X7IwP+CHGil5iypzmwcKZgDpsFTC1FOGCrYsAQqWcrSn0ODHCWA9jzP
5EE4hwcTsakjaeM+ITXSZtuzcjhsQAuEOQQNkCmtLH5c9DQII7KofcaMnkzjF0webv3uEsB9WKpW
93LAcm8kxrieTs2sJDVLnpnKo/bTdhQCzYUc18E0qiZDZoYqGDAlddD7mNEWvEtt3ryjpaeTn4Y5
BGxHTsfPb2YshWwin2k4aZ/8auqWnEvrQmKvW0cAZVMk0NExiXlefxvgb+/yOssDyriaRp8vdKRB
hwEBXyAMREVfVF9pZmlzXzAxf0wOBgkEAH8ABwMBAgJTAYFfJQYABwECAAFfJAYABwECAwFfNziE
```

```
QPdcYC1ZvcWti/oFynQJ/t6QpVLSvKaGAdEivJCVwLrFBFGPxQ0PBK6ijLytRuW4630CKj6T8g==
--trenner--
```

6.9.2 Zweiter Request-Response-Zyklus

```
POST /typo3conf/ext/is_eap/eapGateway.php HTTP/1.1
X-Eid-Authenticationmethod: doStepI
User-Agent: Jakarta Commons-HttpClient/3.1
Host: www.internet-sicherheit.de
Cookie: $Version=0; PHPSESSID=enmkitol5j411fescvq4phjpr1
Content-Length: 1438

MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="trenner"

--trenner
Content-Type: application/octet-stream;
.name="rPicc"
Content-Transfer-Encoding: base64

r7mxG0seeonufRYaD047/M9Gquzyu5UGKTprmSP+qg0qz95OJ0jfECBpJdzy8Y5Pvq5UqCzwiQpc
BfpHgyJW15Gnf1YwirtJ5uZm7R1vIthxGFy6Xg01soeX7einlKIdpQnEiCc7TzisJ+bZSzCERoSy
PVXSVKAOAIVI2gvsvUw=
--trenner
Content-Type: application/octet-stream;
.name="hashOfEphemeralPkPicc"
Content-Transfer-Encoding: base64

AwMXYBoluBaUm8XnEGAMwjxvTIVyD/rDRCG4hw==
--trenner
Content-Type: application/octet-stream;
.name="p"
Content-Transfer-Encoding: base64

ANfBNKomQ2aGKhgwJXXR14ewnwdXl9qJ9X7IwP8=
--trenner
Content-Type: application/octet-stream;
.name="a"
Content-Transfer-Encoding: base64

aKXmLKnObBwpmAOmwVMLUU4YKtiwBCpZytKfQw==
--trenner
Content-Type: application/octet-stream;
.name="b"
Content-Transfer-Encoding: base64

JYD2PM/kQTiHBxOxqSNp4z4hNdJm27NyOGxACw==
--trenner
Content-Type: application/octet-stream;
.name="x"
Content-Transfer-Encoding: base64

DZAprSx+XPQ0CCOyqH3GjJ5M4xdMHm797hLAfQ==
--trenner
Content-Type: application/octet-stream;
.name="y"
Content-Transfer-Encoding: base64

WKpW93LAcm8kxrieTs2sJDVLnpnKo/bTdhQCzQ==
--trenner
Content-Type: application/octet-stream;
.name="n"
Content-Transfer-Encoding: base64

ANfBNKomQ2aGKhgwJXXQ+5jRFrxLbd68o6Wnk58=
--trenner
Content-Type: application/octet-stream;
.name="c"
Content-Transfer-Encoding: base64

AQ==
--trenner--
```

```
HTTP/1.1 200 OK
Date: Sun, 06 Jul 2008 18:17:10 GMT
Server: Apache
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Content-Length: 426
Connection: close
Content-Type: text/html; charset=UTF-8

MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="trenner"

--trenner
Content-Type: application/octet-stream;
.name="HashPkPcdTa"
Content-Transfer-Encoding: base64

OwbayLgboiRoWYmh6QPyzHUkX0/0iP2LGx4KdQ==
--trenner
Content-Type: application/octet-stream;
.name="SignatureBytes"
Content-Transfer-Encoding: base64

MD0CHQCpJLm8xLwOnqoLZ+YXMchlOrQ0sdbHJsrDu1+gAhx1zzDGsVOt82JnRND5YD/1bk4tesVJ
pOrBAFkp
--trenner--
```

6.9.3 Dritter Request-Response-Zyklus

```
POST /typo3conf/ext/is_eap/eapGateway.php HTTP/1.1
X-Eid-Authenticationmethod: doStepN
User-Agent: Jakarta Commons-HttpClient/3.1
Host: www.internet-sicherheit.de
Cookie: $Version=0; PHPSESSID=enmkitol5j411fescvq4phjpr1
Content-Length: 879

MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="trenner"

--trenner
Content-Type: application/octet-stream;
.name="pkPiccKey"
Content-Transfer-Encoding: base64

MIIBEDCB0QYHKoZIzj0CATCBxQIBATAoBgcqhkjOPQEBAh0A18E0qiZDZoYqGDAlddHXh7CfB1eX
2on1fsjA/zA8BBxopeYsqc5sHCmYA6bBUwtRThgq2LAEKlnK0p9DBBwlgPY8z+RBOIcHE7GpI2nj
PiE10mbbs3I4bEALBDkEDZAprSx+XPQ0CCOyqH3GjJ5M4xdMHm797hLAfViqVvdywHJvJMa4nk7N
rCQ1S56ZyqP203YUAs0CHQDXwTSqJkNmhioYMCV10PuY0Ra8S23evKOlp5OfAzoABLlEjZ5JBZDO
0rrMfMKTiBXxBU/SdJtfeAFL3qAKWVWOrBtRnKHzTZ4yxGQy0/5XcywnrFGBZFG8
--trenner
Content-Type: application/octet-stream;
.name="pkPiccSignature"
Content-Transfer-Encoding: base64

MD4CHQDLh4eTTweeTA/2QpkOOEiARwfTGJcXu2Scrmq+Ah0AsWtAWK1RGkNm6UeRMTSEuBNo0+mR
n/xwpBNCog==
--trenner
Content-Type: application/octet-stream;
.name="pkPiccAlgorithm"
Content-Transfer-Encoding: base64

RUNESA==
--trenner--
```

```
HTTP/1.1 200 OK
Date: Sun, 06 Jul 2008 18:17:12 GMT
Server: Apache
Expires: Thu, 19 Nov 1981 08:52:00 GMT
```

```
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Content-Length: 680
Connection: close
Content-Type: text/html; charset=UTF-8

MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="trenner"

--trenner
Content-Type: application/octet-stream;
.name="pkPiccKey"
Content-Transfer-Encoding: base64

MIIBEDCB0QYHKoZIzj0CATCBxQIBATAoBgcqhkjOPQEBAh0A18E0qiZDZoYqGDAlddHXh7CfB1eX
2on1fsjA/zA8BBxopeYsqc5sHCmYA6bBUwtRThgq2LAEKlnK0p9DBBwlgPY8z+RBOIcHE7GpI2nj
PiE10mbbs3I4bEALBDkEDZAprSx+XPQ0CCOyqH3GjJ5M4xdMHm797hLAfViqVvdywHJvJMa4nk7N
rCQ1S56ZyqP203YUAs0CHQDXwTSqJkNmhioYMCV10PuY0Ra8S23evKOlp5OfAzoABDsG2si4G6Ik
aFmJoekD8sx1JF9P9Ij9ixseCnXS7RRDm4EhftNwNb2bDVb9JERMi9NRF2Ts6xy7
--trenner
Content-Type: application/octet-stream;
.name="pkPiccAlgorithm"
Content-Transfer-Encoding: base64

RUNESA==
--trenner--
```

6.9.4 Vierter Request-Response-Zyklus

```
POST /typo3conf/ext/is_eap/eapGateway.php HTTP/1.1
X-Eid-Authenticationmethod: doStepR
User-Agent: Jakarta Commons-HttpClient/3.1
Host: www.internet-sicherheit.de
Cookie: $Version=0; PHPSESSID=enmkitol5j411fescvq4phjpr1
Content-Length: 543

MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="trenner"

--trenner
Content-Type: application/octet-stream;
.name="tPicc"
Content-Transfer-Encoding: base64

gBLGkT1GRynGFfgORZn1og==
--trenner
Content-Type: application/octet-stream;
.name="personalData"
Content-Transfer-Encoding: base64

nBQiSFQd9UUEhpWH968eHxn9OjiT6kQFZ5ojHOWuzCBnytLaXhGE0id7iqxC2xQV1yy4FF/vTspY
OxvdRaSCy3C+OZidGOZN7ysEUIMS
--trenner
Content-Type: application/octet-stream;
.name="iv"
Content-Transfer-Encoding: base64

nJbmIpEBu8NzJ81dRomAmA==
--trenner--
```

```
HTTP/1.1 200 OK
Date: Sun, 06 Jul 2008 18:17:13 GMT
Server: Apache
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
X-Eid-Redirect: https://www.internet-sicherheit.de/authentication-successful/
Content-Length: 0
Connection: close
Content-Type: text/html; charset=UTF-8
```

6.10 Ablauf der Authentisierung aus der Perspektive des Anwenders

Dieser Abschnitt fasst die Authentisierung, insbesondere die Interaktionsschritte aus Sicht des Benutzers zusammen und belegt es mit entsprechenden Abbildungen aus der Beispiel-Implementierung.

6.10.1 Beginn der Web-Authentisierung

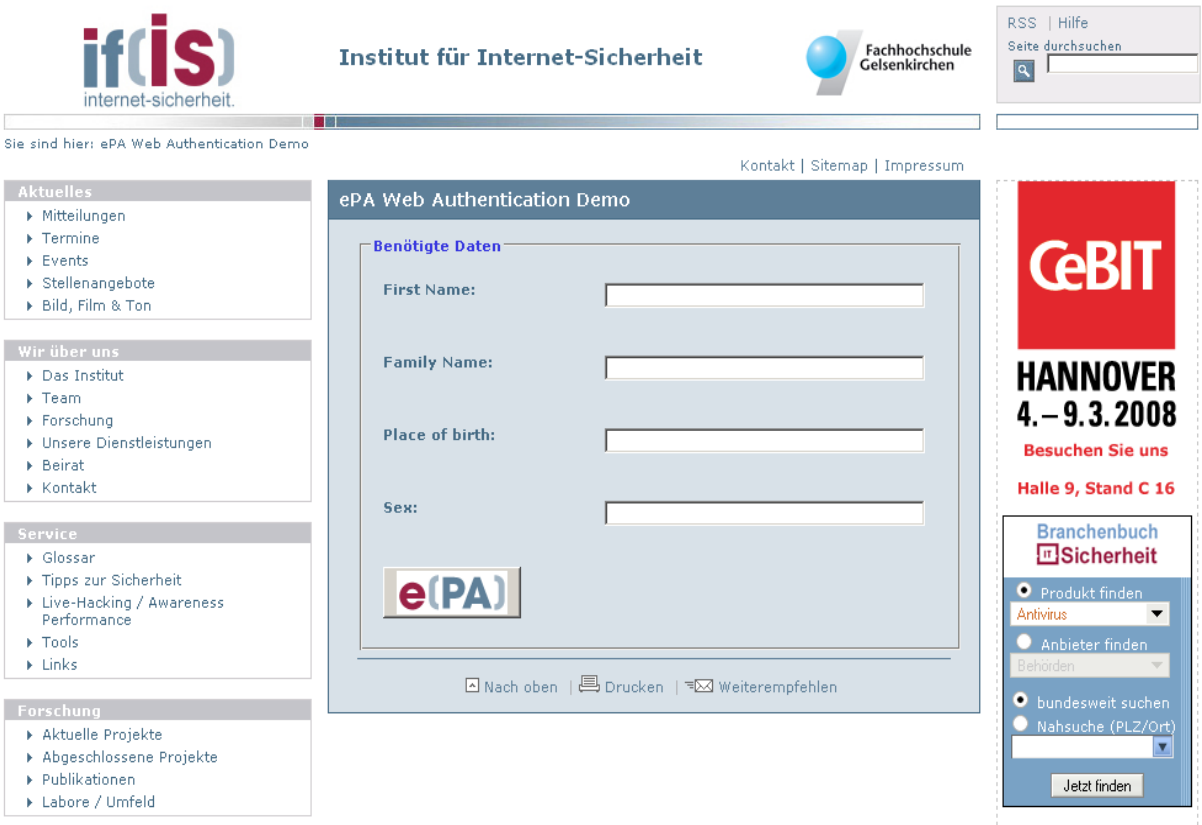


Abbildung 34: Beginn der Web-Authentisierung

Zu Beginn der Web-Authentikation findet der Benutzer ein Formular vor, dessen Eingabefelder üblicherweise manuell gefüllt werden. In dem Beispiel-Szenario können die Felder nicht manuell gefüllt werden – sie sind nur lesbar. Als Alternative benutzt er bei der Web-Authentisierung den ePA-Knopf links unterhalb des Formulars. Dies startet die Authentisierung.

6.10.2 Akzeptieren des Berechtigungszertifikats

Abbildung 35: Der Terminal Certificate Accept Dialog.

Dieser Dialog zeigt den Zertifikatsinhaber der zugreifenden Instanz (Terminal) sowie die Menge der Attribute, auf die zugegriffen werden soll.[16] Der Benutzer hat hier die Möglichkeit, im Sinne der informationellen Selbstbestimmung den Zugriff auf seine personenbezogenen Daten individuell zu erlauben oder zu unterbinden. Ferner hat er die Möglichkeit die gesamte Transaktion zu bestätigen oder abzubrechen.

Im hier dargestellten Beispiel muss der Benutzer darüber entscheiden, ob die Instanz **`DE_T_ifis_01`** auf die personenbezogenen Daten Nachnamen, Vornamen, Geschlecht und Geburtsort zugreifen darf. Das Berechtigungszertifikat ist gültig – dies wird durch die grüne Schriftfarbe der Instanz verdeutlicht. Darüber hinaus kann der Benutzer den Zugriff auf die personenbezogenen Attribute weiter einschränken, indem er die Checkbox vor dem entsprechenden Attribut abwählt.

16 Angaben zur zuständigen Datenschutzaufsichtsbehörde sowie zum Verwendungszweck wurden nicht implementiert, da die Spezifikation keine Felder im Zertifikat für diese Daten vorsieht.

6.10.3 PIN-Eingabe

Abbildung 36: PIN-Eingabe zum Start der Authentisierung

Nachdem der Benutzer den Terminal Certificate Accept Dialog bestätigt hat, muss er seine PIN eingeben, damit ein verschlüsselter Funkkanal zu seinem elektronischen Personalausweis aufgebaut werden kann. Darüber hinaus dient dieser PIN-Eingabe-Dialog der Kenntlichmachung, dass es sich um eine Aktion handelt. Dem Benutzer soll spätestens jetzt bewusst werden, dass nun auf den ePA zugegriffen wird.

6.10.4 Der Authentisierungsprozess

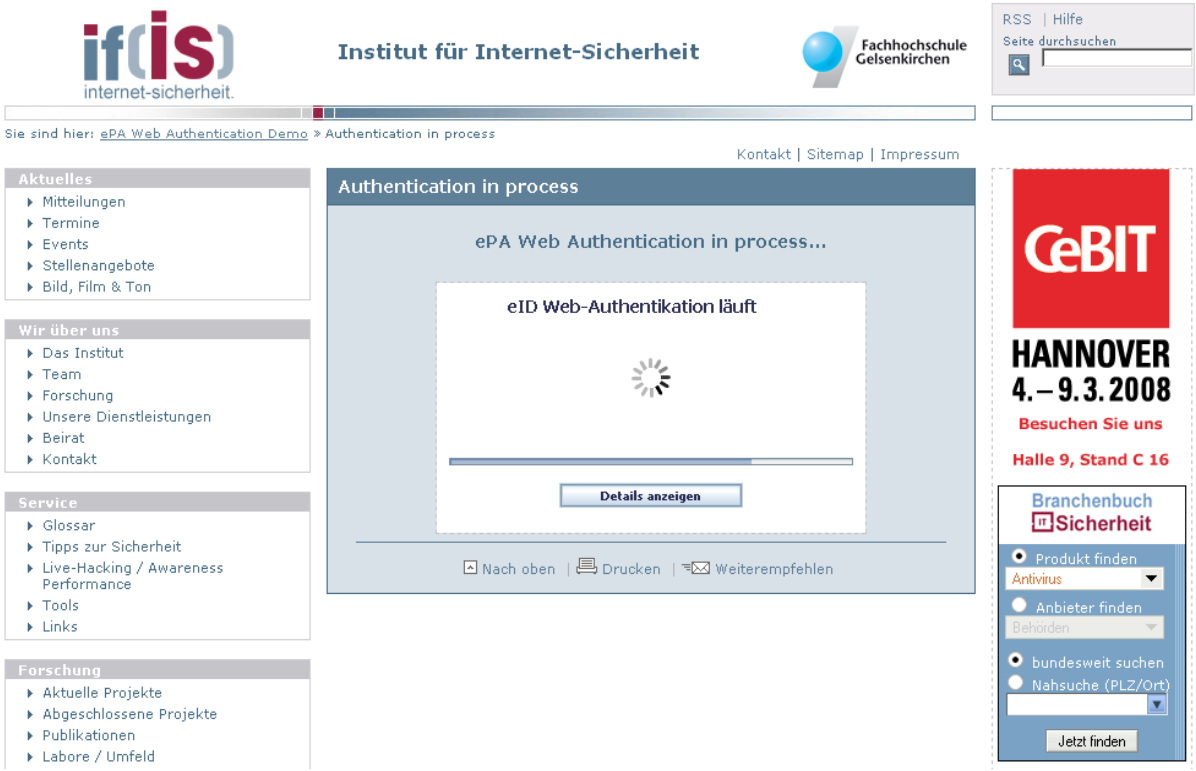


Abbildung 37: Der Authentisierungsprozess ist gestartet

Sofern der Benutzer die korrekte PIN eingegeben hat, startet die Authentisierung. Anhand eines Activity Indicators hat der Benutzer die Möglichkeit die Aktivität des Systems zu überprüfen. Der Fortschrittsbalken zeigt an, zu welchem Anteil der Authentisierungsprozess abgeschlossen ist. Ein Klick auf den Knopf „Details anzeigen“ öffnet ein Logbuch.

6.10.5 Das Logfenster

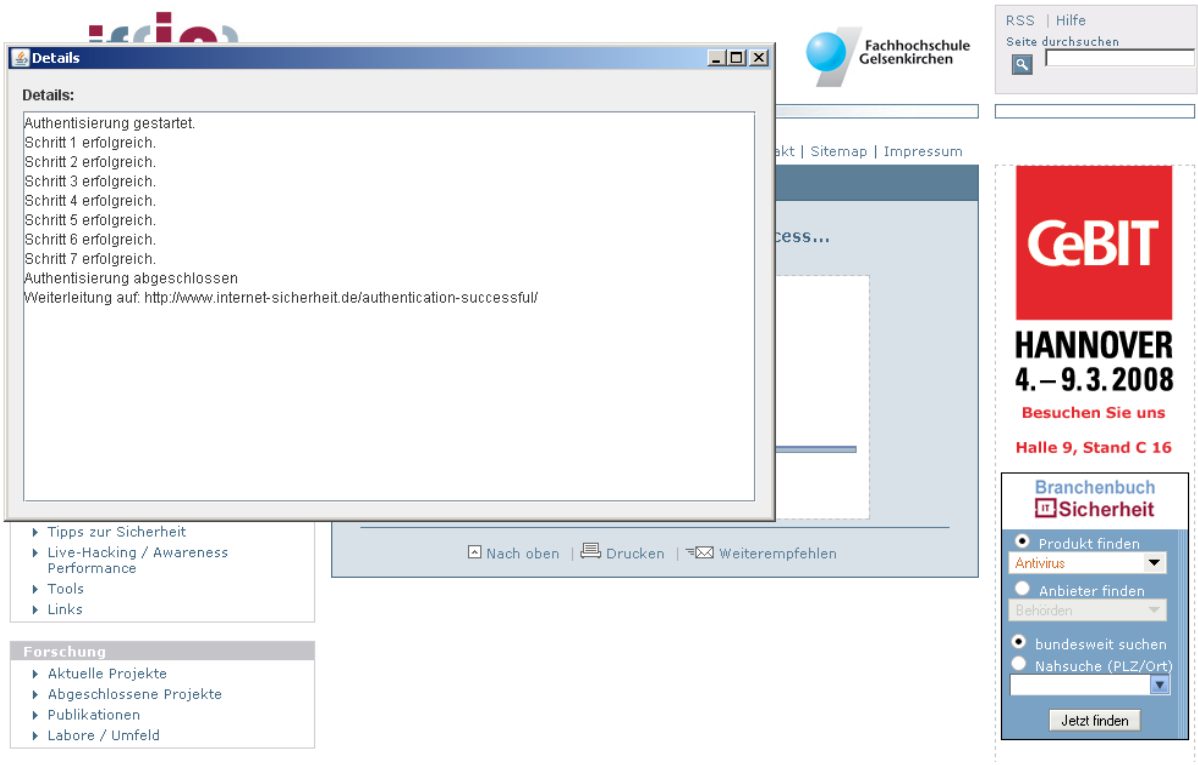


Abbildung 38: Das Logbuch

Der Benutzer hat die Möglichkeit anhand des Logbuchs nachzuvollziehen, welcher Schritt der Authentisierung gerade durchgeführt wird und welche Schritte bereits absolviert sind. Darüber hinaus wird hier die URL, auf die im Erfolgsfall automatisch weitergeleitet wird, dargestellt. Insbesondere technisch versierte Benutzer können den Prozess hier nachvollziehen.
Problematisch ist an dieser Stelle, dass die darzustellenden Schritte noch nicht einheitlich definiert sind. Die hier gewählten 7 Schritte entsprechen in etwa den 4 Request-Response-Zyklen aus Abschnitt 6.9. Damit das Logbuch einen größeren Nutzen bringt, sollten die Schritte genau definiert werden und implementierungsübergreifend einheitlich kommuniziert werden.

6.10.6 Authentisierung erfolgreich abgeschlossen

Abbildung 39: Authentisierung erfolgreich

Nach erfolgreicher Authentisierung sind die Attributswerte auf Serverseite verfügbar und können beispielsweise als Formularinhalte verwendet werden. Hier werden die Daten dem Benutzer angezeigt. Die Authentisierung ist damit erfolgreich abgeschlossen.

7 Fazit und Ausblick

Mit der Einführung des elektronischen Personalausweise erwartet uns eine große Neuerung mit unbekannter Tragweite. Auf der einen Seite werden die bisher ausschließlich physikalischen Sicherheitsmerkmale ergänzt durch elektronische, insbesondere unter Verwendung von Biometrie – vor dem Hintergrund der hohen Datenschutzanforderungen in Deutschland ist dies sicherlich eine Herausforderung. Die elektronischen Sicherheitsfunktionen werden helfen, um die Echtheit und die Zugehörigkeit des Trägers des Ausweisdokuments zum Eigentümer besser feststellen zu können.
Auf der anderen Seite benötigen wir verlässliche Mechanismen, um Identitätsprüfungen auch über das Internet abwickeln zu können. Dies ist Voraussetzung dafür, dass bestimmte, bisher nur in Verbindung mit einer Offline-Identitätsprüfung nutzbaren Dienste, bald auch online eingesetzt werden können. Der elektronische Personalausweis schafft damit eine Basis, sowohl für eBusiness als auch eGovernment-Dienste.

Mit der Spezifikation von Extended Access Control ist aus technischer Perspektive ein bedeutender Meilenstein geschafft, um eine sichere Methode der Übertragung von schützenswerten Daten über das Internet zu etablieren. Diese Arbeit zeigt zum ersten Mal, dass EAC auch für den elektronischen Personalausweis implementierbar ist und sie weist auf wichtige Implementierungsdetails hin. Der Kern der Arbeit besteht darin, die formale Spezifikation von EAC auf das Internet, konkreter auf Web-Anwendungen, zu übertragen. Sie macht deutlich, dass zum derzeitigen Stand nicht alle Komponenten gemäß den Forderungen des Datenschutzes implementiert werden können. So fordern die Juristen beispielsweise die verlässliche Angabe der zuständigen Datenschutzaufsichtsbehörde sowie des Verwendungszwecks, allerdings sieht die Spezifikation keine entsprechenden Felder für diese Daten vor. Summa summarum ist diese Arbeit eine wichtige Grundlage für zukünftige Implementierungen. Für die Zukunft können die simulierten Komponenten dieser Beispiel-Implementierung nach und nach durch echte ersetzt werden. Ein erster Schritt wäre etwa die Einbindung eines Hochsicherheitsmoduls im

EAS oder die Anbindung einer (RF-)SmartCard als ePA.

Ohne Zweifel – es wird meiner Meinung nach auch mit der Online-Authentikationsfunktion beim elektronischen Personalausweis Probleme geben. In einer Zeit, in der Botnetze aus bis zu einer 7-stelligen Zahl an gekaperten Computern bestehen (vgl. [Sterling 1.5M botnet]) und möglicherweise bis zu 25% aller Computer mit Internetanschluss infiziert sind (vgl. [Heise Botnet Cerf]), ist es nur eine Frage der Zeit, bis dies auch im Zusammenhang mit dem elektronischen Personalausweis aneckt. Erst wenn Technologien wie Trusted Computing Fuß gefasst haben und die Software auf dem Client in einer sicheren Umgebung läuft, könnte auch diese Gefahr minimiert werden.

Insbesondere auf nationaler Ebene hat der elektronische Personalausweis aufgrund seiner hohen Verbreitung durch die Ausweispflicht einen enormen Vorteil. Heute kann noch keine Aussage darüber getroffen werden, wer die Early Adopters der Authentikationsfunktion sein werden, aber meiner Meinung nach birgt sie großes Potential, um irgendwann einen Stammplatz in der Reihe der Authentikationsfunktionen einzunehmen.

Anhang

Anmerkungen zur BSI TR 03110 in der Version 2.0

Bei der Durchsicht der BSI TR 03110 wurden folgende Verbesserungs-vorschläge ausgearbeitet.

Nummerierung der Algorithmen-Schritte

Um eine eindeutige Identifikation eines bestimmten Schritts innerhalb eines Algorithmus bzw. Protokoll-Ablaufs zu ermöglichen, sollten jegliche Schritte der Protokollbeschreibungen nummeriert werden. Dies betrifft insbesondere die Protokollbeschreibungen des PACE, der Chip Authentication sowie der Terminal Authentication in den Abschnitten 4.2, 4.3 und 4.4.

Weitere Vorschläge

Das Dokument BSI TR 03110 beschreibt in erster Linie die Anwendung der erweiterten Zugangskontrollmechanismen (Extended Access Control) für die Anwendung des elektronischen Reisepasses (ePass). Die Mechanismen eignen sich gleichwohl auch zur Anwendung beim elektronischen Personalausweis. Allerdings werden die Protokolle in unterschiedlicher Reihenfolge ausgeführt. Während beim ePass zunächst die Chip Authentication und danach die Terminal Authentication durchgeführt wird, beginnt man bei der eID-Anwendung mit der Terminal Authentication und anschließend folgt die Chip Authentication.

Dies äußert sich beispielsweise darin, dass im Rahmen der Chip Authentication im Abschnitt 4.3.1 der [BSI-TR-03110] in Schritt 3.c) der Hashwert des flüchtigen, öffentlichen Schlüssels zur Vorbereitung der Terminal Authentication berechnet wird. Bei der geänderten Reihenfolge im Rahmen der eID-Anwendung ist dies an dieser Stelle nicht mehr nötig, da die Terminal Authentication bereits abgeschlossen ist. Vielmehr muss dieser Schritt bereits vor der Terminal Authentication ausgeführt werden.

Ein ähnliches Phänomen ist die Tatsache, dass in den Protokollbeschreibungen zu Beginn der Terminal Authentication die Domain Parameter nicht zwischen

beiden Seiten ausgetauscht werden, sondern nur in der Chip Authentication. Wenn – wie bei der eID-Anwendung der Fall – die Terminal Authentication vor der Chip Authentication ausgeführt wird, fehlen die Domain Parameter evtl. auf einer der beiden Seiten (z.B. auf der Seite des Terminals in Abschnitt 4.4.1 Schritt 3, erster Teil). Daher sollte zu Beginn der Terminal Authentication ein Schritt eingefügt werden, in dem die Domain Parameter übertragen wird.

Im Rahmen der Terminal Authentication wird der Hashwert des flüchtigen, öffentlichen Schlüssels **~PK_PCD_TA** an den MRTD Chip übertragen. Dies wird auf Seite 31 angedeutet durch

> „The terminal MAY send this hash as part of Terminal Authentication to indicate that Chip Authentication will be performed after Terminal Authentication.“ (S. 31, [BSI-TR-03110])

An dieser Stelle muss erwähnt werden, dass später im Rahmen der Chip Authentication geprüft werden muss, ob der in 3.a) genutzte flüchtige, öffentliche Schlüssel den hier übermittelten Hashwert besitzt. Dies ist in der Protokollbeschreibung nicht erwähnt. Andererseits könnte man bei der Reihenfolge Terminal Authentication → Chip Authentication anstelle der Übermittlung des Hashwerts direkt den passenden flüchtigen, öffentlichen Schlüssel übertragen, da dieser im Rahmen der Chip Authentication benötigt wird.
Im Rahmen der Chip Authentication in Abschnitt 4.3.1 der [BSI-TR-03110] muss **PK_PICC** im ersten Schritt in signierter Form (z.B. durch einen Document Signer) übertragen werden.

Literaturverzeichnis

[Apache httpclient]: Apache Software Foundation, Apache Jakarta Commons HTTP Client for HTTP/1.0 and HTTP/1.1, 2007, http://hc.apache.org/httpclient-3.x/

[Beel und Gipp 2005]: Jöran Beel, Bela Gipp, ePass - der neue biometrische Reisepass, 2005, Shaker Verlag Aachen

[BelgRootCA]: Belgium Root CA, Root CA Zertifikate und Sperrlisten, 2008, http://repository.eid.belgium.be/DE/RootCA.htm

[BMI 2004 Sep11]: Bundesministerium des Innern, Nach dem 11.September 2001 - Maßnahmen gegen den Terror - Dokumentation aus dem Bundesministerium des Innern, 2004, http://www.bmi.bund.de/Internet/Content/Common/Anlagen/Broschueren/2004/Nach__dem__11__September__2001__Massnahmen__Id__25583__de,templateId=raw,property=publicationFile.pdf/Nach_dem_11_September_2001_Massnahmen_Id_25583_de.pdf

[BMI Fälschungssicherheit]: Bundesministerium des Innern, Pässe noch fälschungssicherer, 2004, http://www.bmi.bund.de/cln_012/nn_122688/sid_E9E44005F71B90E1D8211E8469FF4818/Internet/Content/Themen/Kriminalitaet/Einzelseiten/Paesse__noch__faelschungssicherer__Id__62490__de.html

[BMI Grobkonzept ePA]: Bundesministerium des Innern, IT4, Einführung des elektronischen Personalausweises in Deutschland - Grobkonzept Version 2.0, 2008, http://asset.netzpolitik.org/wp-upload/bmi_epa-grobkonzept-2-0_2008-07-02.pdf

[Boundycastle Java API]: Bouncycastle.org, Welcome to the home of the Legion of the Bouncy Castle Java cryptography APIs., 2008, http://www.bouncycastle.org/java.html

[BrainpoolECC]: ECC Brainpool, ECC Brainpool Standard Curves and Curve Generation, 2005

[BReg 2005]: Bundesregierung, Antwort der Bundesregierung auf die Kleine Anfrage der Abgeordneten Gisela Piltz, Ulrike Flach, Rainer Funke und weiterer vom Januar 2005, 2005, http://dip.bundestag.de/btd/15/046/1504616.pdf

[BReg 2007]: Bundesregierung, Antwort der Bundesregierung auf die Kleine Anfrage der Abgeordneten Jan Korte, Ulla Jelpke, Petra Pau und der Fraktion DIE LINKE vom November 2007, 2007, http://dip.bundestag.de/btd/16/070/1607073.pdf

[BSI-TR-03110]: BSI, Advanced Security Mechanisms for Machine Readable Travel Documents - Extended Access Control (EAC), 2007

[BSI-TR-03111]: BSI, Elliptic Curve Cryptography based on ISO 15946, 2007

[BSI-TR-03116]: BSI, Technische Richtlinie für die eCard-Projekte der Bundesregierung, 2007

[BSI-TR-20102]: BSI, Kryptographische Verfahren: Empfehlungen und Schlüssellängen, 2007

[DP PostIdent]: Deutsche Post AG, Identitätsprüfung - denn sicher ist sicher (PostIdent-Verfahren der Deutschen Post AG), 2008, http://www.deutschepost.de/dpag?tab=1&skin=hi&check=no&lang=de_DE&xmlFile=1015469

[ELSTER 2008]: ELSTER, ELSTER Webseite: https://www.elsteronline.de/eportal/, 2008, https://www.elsteronline.de/eportal/

[EU/EG Nr.2252/2004]: Europäische Union, Verordnung (EG) Nr. 2252/2004 des Rates vom 13.Dezember 2004 über Normen für Sicherheitsmerkmale und biometrische Daten in von den Mitgliedstaaten ausgestellten Pässen und Reisedokumenten, 2004, http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2004:385:0001:0006:DE:PDF

[FlexSecure CVREF]: FlexSecure/BSI, Official CVCA Demonstrator Website for the ePassport interoparability tests, 2007, http://flexsecure.eu/cvca-demo/index.html

[GoF 1995]: Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, Design Patterns. Elements of Reusable Object-Oriented Software, 1995, Addison Wesley

[Heise BelgPass]: Daniel Bachfeld, Belgische ePässe ohne Autorisierung auslesbar, 2007,

http://www.heise.de/newsticker/Belgische-ePaesse-ohne-Autorisierung-auslesbar--/meldung/90962

[Heise Botnet Cerf]: Daniel Bachfeld, Vint Cerf: Ein Viertel der Internet-PCs ist Mitglied eines Bot-Netzes, 2007, http://www.heise.de/newsticker/Vint-Cerf-Ein-Viertel-der-Internet-PCs-ist-Mitglied-eines-Bot-Netzes--/meldung/84317

[Heise HollPass]: Andreas Wilkens, ePass-Hack im niederländischen TV demonstriert, 2006, http://www.heise.de/security/ePass-Hack-im-niederlaendischen-TV-demonstriert--/news/meldung/69127

[ICAO 2006]: International Civil Aviation Organization, Doc 7300/9 Convention on International Civial Aviation (dt.: Konvention zum internationalen zivilen Flugverkehr), 2006, http://www.icao.int/icaonet/arch/doc/7300/7300_9ed.pdf

[ICAO-BIOM-MRTD03]: Secretariat Facilitation Division, Biometric technology in machine readable travel documents - the ICAO blueprint, 2003 ,

[ICAO-MRTD-LDS]: Secretariat Facilitation Division, Technical Report - Development of a logical data structure LDS for optional capacity expansion technologies, 2004, http://www.mrtd.icao.int/images/stories/Doc/ePassports/Logical%ac%ac_Data_Structure(LDS)_version1.7.pdf

[ICAO-PKI-MRTD-ICC]: Tom A.F. Kinneging (ICAO), PKI for Machine Readable Travel Documents offering ICC Read-Only Access, 2004

[ISO/IEC 14443]: ISO/IEC, Identification cards -- Contactless integrated circuit(s) cards -- Proximity cards, 2008

[ISO/IEC 7810]: ISO/IEC, Identification cards - Physical characteristics, 2005

[ISO/IEC 7816-4]: ISO/IEC, Identification cards - Integrated circuit cards - Part 4: Organization, security and commands for interchange, 2005

[JavaJarsigner]: Sun Microsystems, jarsigner - JAR Signing and Verification Tool, 2004, http://java.sun.com/j2se/1.5.0/docs/tooldocs/solaris/jarsigner.html

[JavaKeytool]: Sun Microsystems, keytool - Key and Certificate Management Tool, 2002, http://java.sun.com/j2se/1.4.2/docs/tooldocs/windows/keytool.html

[JavaSecArch]: Sun Microsystems, Java 2 Platform Security Architecture, 2002, http://java.sun.com/j2se/1.4.2/docs/guide/security/spec/security-spec.doc.html

[Kuegler und Naumann 2007]: Dennis Kuegler, Ingo Naumann, Sicherheitsmechanismen für kontaktlose Chips im deutschen Reisepass, 2007 Datenschutz und Datensicherheit 31 , vieweg

[PECL_HTTP]: Michael Wallner, pecl_http: Extended HTTP support for PHP, 2008, http://pecl.php.net/package/pecl_http

[PHP]: The PHP Group, PHP: Hypertext Preprocessor, 2008, http://de.php.net/

[Reisen 2008]: Andreas Reisen, Digitale Identität im Scheckkartenformat - Datenschutzvorkehrungen für den elektronischen Personalausweis, 2008 Datenschutz und Datensicherheit 3/2008, vieweg

[RFC 2045]: N. Freed, N. Borenstein, Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies, 1996

[RFC 2046]: N. Freed, N. Borenstein, Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types, 1996

[RFC 2047]: K. Moore, MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text, 1996

[RFC 2049]: N. Freed, N. Borenstein, Multipurpose Internet Mail Extensions (MIME) Part Five: Conformance Criteria and Examples, 1996

[RFC 2387]: E. Levinson, The MIME Multipart/Related Content-type, 1998

[RFC 2616]: R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach et al., Hypertext Transfer Protocol -- HTTP/1.1, 1999

[RFC 2663]: P. Srisuresh, M. Holdrege, IP Network Address Translator (NAT) Terminology and Considerations, 1999

[RFC 2817]: R. Khare, S. Lawrence, Upgrading to TLS Within HTTP/1.1, 2000

[RFC 2818]: E. Rescorla, HTTP Over TLS, 2000

[RFC 4288]: N. Freed, J. Klensin, Media Type Specifications and Registration Procedures, 2005

[RFC 4289]: N. Freed, J. Klensin, Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures, 2005

[Roßnagel et. al 2008]: Alexander Roßnagel, Gerrit Hornung, Christoph Schnabel, Die Authentisierungsfunktion des elektronischen Personalausweises aus datenschutzrechtlicher Sicht, 2008 Datenschutz und Datensicherheit , vieweg

[Schmidt 2006]: Jürgen Schmidt, The hole trick - How Skype & Co. get round firewalls, 2006, http://www.heise-online.co.uk/security/How-Skype-Co-get-round-firewalls--/features/82481

[Sterling 1.5M botnet]: Toby Sterling, Dutch say suspects hacked 1.5 million computers, 2005, http://www.usatoday.com/tech/news/computersecurity/2005-10-20-dutch-hack_x.htm

[Typo3]: Typo3 Association, typo3.org: TYPO3 Content Management System - Developer Resource, 2008, http://typo3.org/

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

Abkürzung	Erläuterung
APDU	Application Protocol Data Unit
ASN.1	Abstract Syntax Notation 1
BDSG	Bundesdatenschutzgesetz
BMI	Bundesministerium des Innern
BSI	Bundesamt für Sicherheit in der Informationstechnik
CA	Certificate Authority
CMS	Content Management System
CSCA	Country Signing Certificate Authority
CVCA	Country Verifying Certificate Authority
DoS	Denial of Service
DS	Document Signer
DSA	Digital Signature Algorithm
DV	Document Verifier
EAC	Extended Access Control
ECC	Elliptic Curve Cryptography
ECDH	Elliptic Curve Diffie-Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
eID	Electronic ID, elektronischer Personalausweis
EKK	Elliptische-Kurven-Kryptographie (Kryptographie der elliptischen Kurven)
ePA	Elektronischer Personalausweis
ePass	Elektronischer Reisepass
HTTP	Hypertext Transfer Protocol
ICAO	International Civil Aviation Organization
MAC	Message Authentication Code
MIME	Multipurpose Internet Mail Extensions
MITM	Man in the middle
MRTD	Machine Readable Travel Document
MRZ	Maschinenlesbare Zone, machine readable zone
NAT	Network Address Translation
nPA	Neuer elektronischer Personalausweis
OID	Object Identifier
PACE	Password Authenticated Connection Establishment
RSA	Rivest Shamir Adleman (Kryptosystem)
SSL	Secure Sockets Layer
TC	Terminal Certificate
TMG	Telemediengesetz
TLS	Transport Layer Security
TR	Technische Richtlinie
WWW	World Wide Web