Diplomarbeit

Benutzerkonten-Verwaltung
mit dem neuen Personalausweis

Mathias Jeschke

Berlin, den 1. Juni 2011
Inhaltsverzeichnis

1. Einleitung 1
   1.1. Motivation ........................................ 1
   1.2. Zielstellung ...................................... 1
   1.3. Aufbau der Arbeit ................................. 2

2. Authentifikationsverfahren 3
   2.1. Begriffe .......................................... 3
       2.1.1. IT-System .................................. 3
       2.1.2. Schutzziele ................................ 3
       2.1.3. Digitale Identität ............................. 4
       2.1.4. Authentifikation ............................... 5
       2.1.5. Autorisierung ................................ 6
   2.2. Authentifikationsverfahren – Ein Überblick ........ 6
       2.2.1. Wissenbasierte Verfahren ...................... 6
       2.2.2. Besitzbasierte Verfahren ...................... 7
       2.2.3. Biometrische Verfahren ....................... 8
   2.3. Passwortbasierte Authentifikationsverfahren .......... 9
       2.3.1. Prinzipielle Probleme und Gegenmaßnahmen ... 9
   2.4. Alternative Authentifikationsverfahren .......... 12
       2.4.1. Einmalpasswörter ................................ 12
       2.4.2. Challenge-Response-Verfahren .................. 14
   2.5. Passwortrücksetzverfahren .......................... 15
       2.5.1. Die PUK – Super-Passwort für dezentrale Entsperrung 16
       2.5.2. Zentrale Maßnahmen ............................. 16

3. Die digitale Identität mit dem neuen Personalausweis 19
   3.1. Der neue Personalausweis .......................... 20
       3.1.1. Authentisierung versus Signatur .................. 20
       3.1.2. Physische Sicht .................................. 21
       3.1.3. Logische Sicht: Anwendungen ..................... 22
       3.1.4. Zugriff auf die Ausweisdaten ..................... 22
       3.1.5. Berechtigungszertifikate ......................... 23
   3.2. Die eID-Anwendung .................................. 23
       3.2.1. Komponenten ...................................... 24
       3.2.2. Ablauf einer eID-Sitzung .......................... 25
       3.2.3. Sicherheit und Datenschutz ....................... 27
   3.3. eID-Szenarien ..................................... 30
       3.3.1. Komponenten ...................................... 31
# Inhaltsverzeichnis

3.3.2. eID-Server ................................................................. 32  
3.3.3. Anforderungen an die eID-Infrastruktur .......................... 34  
3.3.4. Szenario I: Lokaler eID-Server ................................... 34  
3.3.5. Szenario II: Entfernter eID-Server mit Vollzugriff ............... 35  
3.3.6. Anwendungstest: eID-Service der Bundesdruckerei ............... 35  
3.3.7. Szenario III: Entfernter eID-Server mit eingeschränktem Zugriff 39  
3.3.8. Fazit ............................................................................ 40  

4. Entwurf eines Benutzeroberungsdienstes .............................. 41  
4.1. Zielstellung ....................................................................... 41  
4.2. Fragestellungen .................................................................. 41  
4.3. Identitätsmanagement ....................................................... 42  
4.3.1. Ist-Zustand .................................................................... 42  
4.3.2. Speicherung der Personendaten ...................................... 44  
4.4. Prozesse ............................................................................. 46  
4.4.1. Zurücksetzen des Account-Passworts .............................. 46  
4.4.2. Verknüpfen eines bestehenden Accounts mit einem neuen Ausweis 47  
4.4.3. Anlegen eines neuen Accounts ....................................... 48  
4.5. Entwurfsentscheidungen .................................................... 49  
4.5.1. Umsetzung der Funktionen im Prototyp ............................ 50  
4.6. Architektur ......................................................................... 51  
4.6.1. Webserver ...................................................................... 52  
4.6.2. Anwendungsserver ....................................................... 52  
4.6.3. Verzeichnisdienst .......................................................... 52  
4.6.4. eID-Server .................................................................... 53  
4.6.5. Testlogin-System .......................................................... 53  
4.6.6. Anwendungsmodul ........................................................ 53  
4.7. Datenschutzrechtliche Betrachtungen ................................ 55  
4.7.1. Betroffene Datengruppen .............................................. 55  
4.7.2. Datenerhebung und -speicherung ..................................... 56  
4.7.3. Datenzugriff ................................................................. 56  
4.8. Sicherheit der Dienstanwendung ........................................ 56  

5. Implementierung .................................................................. 59  
5.1. Abhängigkeiten und Auswahlkriterien ............................... 59  
5.2. Komponenten des Dienstes und Werkzeuge ......................... 60  
5.2.1. Virtuelle Maschinen zum Aufbau der Demonstrationsumgebung 60  
5.2.2. VM I – Server-Dämonen ............................................... 60  
5.2.3. VM II – Testlogins-System ............................................. 62  
5.2.4. Testclient ...................................................................... 63  
5.2.5. Entwickler-PC .............................................................. 63  
5.3. Die Dienstanwendung ....................................................... 64  
5.3.1. Verzeichnisdienst .......................................................... 64  
5.3.2. Module .......................................................................... 64  
5.3.3. eID-Anbindung ............................................................. 65  
5.3.4. Konfiguration ............................................................... 65
5.3.5. Besonderheiten bei der Implementierung .......................... 66
5.3.6. Quelltexte .................................................. 67

6. Fazit 69
  6.1. Zusammenfassung ............................................. 69
  6.2. Ausblick ...................................................... 70

A. Screenshots der Webanwendung 71
  A.1. Startseite: Modulauswahl ................................. 71
  A.2. Test-Modul .................................................. 72
  A.3. Ein neues Passwort setzen ............................... 73
  A.4. Einen neuen Account beantragen ......................... 75
  A.5. Fehlermeldung .............................................. 77

B. Konfigurationsdateien 79
  B.1. LDAP ......................................................... 79
  B.2. Benutzerverwaltungsdienst ............................... 81

Abbildungsverzeichnis 85
Abkürzungsverzeichnis 87
Literaturverzeichnis 89
1. Einleitung

1.1. Motivation

Heutzutage werden Geschäftsprozesse und Dienstleistungen zunehmend über das Internet abgewickelt. Dabei spielt die Identifizierung unbekannter Geschäftspartner und Benutzer sowie die Authentifizierung bzw. Wiedererkennung bekannter Geschäftspartner und Benutzer eine wichtige Rolle.

Mit dem neuen Personalausweis (nPA) gibt es erstmals in Deutschland ein elektronisches Identitätsdokument, dass zum einen einer breiten Anwenderschaft zur Verfügung steht und zum anderen durch eine hoheitliche und damit sehr vertrauenswürdige Institution – der Bundesrepublik Deutschland – ausgestellt wird. Mit der eID-Funktion des nPA wird somit einerseits die Identifizierung im elektronischen Handel (E-Commerce) und elektronischen Verwaltung (E-Governement) ermöglicht und andererseits die eindeutige Wiedererkennung eines Ausweisinhabers gewährleistet.


1.2. Zielstellung

Die vorliegende Arbeit soll aufzeigen, inwiefern der neue Personalausweis die Verwaltung von Benutzerkonten bei einer Organisation unterstützen kann. Es ist zu prüfen, welche Möglichkeiten der Anbindung an einen eID-Service bestehen, um die Daten und Funktionen von Testausweisen im Rahmen des offenen Anwendungstests des neuen Personalausweises nutzen zu können.

In einem Software-Prototyp sollen typische Anwendungsfälle realisiert und demonstriert werden. Dabei spielen Fragen nach der Sicherheit des zu entwerfenden Benutzerverwaltungsdienstes eine wichtige Rolle.

\[\text{Den bisherigen „alten“ Personalausweis besitzen mehr als 60 Mio. Bundesbürger, vgl.:}\]
\[\text{http://bmi.bund.de/SharedDocs/FAQs/DE/Themen/Sicherheit/epersonalausweis_faq_chip.html}\]
1. Einleitung

1.3. Aufbau der Arbeit

In Kapitel 2 werden mögliche Authentifikationsverfahren vorgestellt. Es wird dargestellt, welche Probleme im Zusammenhang mit Passwortverfahren auftreten und welche Lösungen existieren. Auf alternative Verfahren wird hingewiesen.

Kapitel 3 widmet sich dem neuen Personalausweis und untersucht die damit möglichen eID-Szenarien zur Nutzung des Ausweises.

Darauf aufbauend wird in Kapitel 4 ein Dienst zur Benutzerverwaltung modelliert, wie er hauptsächlich im universitären Umfeld, aber auch in großen Organisationen zum Einsatz kommen könnte.

Schließlich gibt Kapitel 5 einen Überblick über die praktische Realisierung eines Prototyps und skizziert die Struktur des Dienstes aus Softwaresicht.

Im letzten Kapitel wird die Arbeit durch eine Zusammenfassung und einen Ausblick abgeschlossen.
2. Authentifikationsverfahren

Trotz enormer Fortschritte auf dem Gebiet der IT-Sicherheit und den damit verfügbaren, sehr sicheren Verfahren ist das Passwort in vielen Fällen auch heutzutage immer noch der Schlüssel zu virtuellen Welten, wie z. B. dem Internet und den dort angebotenen elektronischen Diensten.

In diesem Kapitel werden zunächst existierende Authentifikationsverfahren vorgestellt, insbesondere solche, die auf einem Passwort basieren. Anschließend werden Möglichkeiten erläutert, wie ein Passwort zurückgesetzt werden kann.

2.1. Begriffe


2.1.1. IT-System

Ein IT-System (im Weiteren einfach: „System“) ist eine technische Einrichtung, die für ein Subjekt Daten speichert und verarbeitet. Im Fokus dieses Kapitels stehen dabei (Teil-)Systeme die sicherstellen, dass ausschließlich berechtigte, also authentifizierte Subjekte eine Datenverarbeitung veranlassen können.

2.1.2. Schutzziele

In einem System gewährleistet die Vertraulichkeit, dass keine unautorisierte Informationsgewinnung möglich ist, d. h., es ist sichergestellt, dass sensible Daten und Nachrichten unkenntlich gemacht bzw. verborgen werden. In der Praxis wird dies meist durch Verschlüsselung der Daten erreicht, vgl. [Eck08b], S. 8.

Die Integrität (auch Unverfälschtheit genannt) garantiert, dass eine unzulässige Veränderung von Daten festgestellt werden kann.

Zur Überprüfung der Glaubwürdigkeit von Subjekten und Objekten anhand von charakteristischen Eigenschaften eines Systems wird als Authentizität (des Subjekts bzw. Objekts) bezeichnet. Konkret erfolgt diese Überprüfung mit Hilfe von Identitäten, die durch eine Authentifikation erzeugt werden. Die Authentizität von Objekten dagegen wird durch einen Urheber- bzw. Ursprungsachweis gewährleistet, was vorrangig bei verteilten Systemen notwendig ist und mit kryptographischen Verfahren umgesetzt wird ([Eck08b], S. 7). Im Vergleich zur Authentizität von Objekten (in Form von Nachrichten) ermöglicht die Verbindlichkeit, dass nicht nur der eigentliche Empfänger, sondern auch Dritte die Unverfälschtheit und Urheberschaft feststellen können ([BNS05], S. 2).
2. Authentifikationsverfahren


Anonymität und Pseudonymität geben an, ob sich eine Identität gar nicht bzw. nur eingeschränkt zurückverfolgen lässt. Diese Schutzziele sind meist direkt aus Gesetzen und Bestimmungen, wie dem Datenschutz, abgeleitet.

In konkreten Systemen werden einzelne Schutzziele oftmals geeignet kombiniert, um die spezifischen Schwächen zu verringern und ihre Stärken zu vereinen. So hilft eine vertrauliche Kommunikation durch Verschlüsselung der Daten nur gegen passive Angreifer. Aktive Angreifer dagegen können mittels eines Man-in-the-Middle-Angriffs (MITM) durch separate Kanäle eine vertrauliche (weil verschlüsselte) Kommunikation vorgaukeln. Daher wird häufig eine gegenseitige Authentifikation beider Kommunikationspartner angestrebt.\(^1\)

2.1.3. Digitale Identität


---


\(^2\)In [Win05], S. 9 werden diese Daten nach Attributen, Vorlieben (engl. preferences) und Merkmalen (engl. traits) unterschieden. Merkmale sind in diesem Kontext als Identifikatoren besonders geeignet.

\(^3\)Dieses Beispiel zeigt auch deutlich: Ein Entifikator allein kann unter Umständen nicht zur Bestimmung der Entität ausreichen – so lässt sich die MAC-Adresse mancher Computer, ebenso wie der Fingerabdruck einer Person manipulieren.
In IT-Systemen dienen (digitale) Identitäten heutzutage als Grundlage vieler Prozesse. Mit ihnen lässt sich die Zuordnung von Daten, die Abrechnung von genutzten Diensten und die Zugriffskontrolle von Objekten innerhalb des Systems sinnvoll umsetzen.

Digitale Identitäten werden in Form von Benutzerkonten Gegenstand weiterer Betrachtungen in dieser Arbeit sein.


Das folgende Zitat von Mario Grobholz beschreibt den Zusammenhang zwischen Identitäten und Personen:


2.1.4. Authentifikation


6So legen Browserhersteller fest, welche Root-CAs in der Standardeinstellung als vertrauenswürdig gelten.
In einem (Ausweis-)Dokument werden die charakteristischen Identitätsattribute des Provers, welche demzufolge vom Issuer sorgfältig zu prüfen sind, an eine Menge von Authentifikatoren (z. B. ein Lichtbild, Fingerabdrücke oder einen öffentlichen Schlüssel) gebunden. Wichtig ist dabei, dass der Verifier in der Lage sein muss, die Authentizität des Dokuments zu prüfen. In Ausweisdokumenten geschieht das durch Sicherheitsmerkmale; in Zertifikaten durch eine digitale Signatur, vgl. [Mü10], S. 46ff.

Die Art und Weise zur Erbringung des Identitätsnachweises, wird durch Authentifikationstechniken charakterisiert. Diese beruhen auf Faktoren dieser drei Kategorien:

- Wissen (etwas, das jemand weiß),
- Besitz (etwas, das jemand besitzt),
- Biometrie (etwas, das jemand ist).


**Anforderungen an Authentifikationsverfahren**

Bei der Bewertung von Authentifikationsverfahren helfen folgende Anforderungen (vgl. [BNS05], S. 217f.):

- „Durchführbarkeit: Ein berechtigter Prover muss einem Verifier gegenüber in der Lage sein, seine Identität nachzuweisen."
- Keine Impersonation: Kein Teilnehmer eines Systems darf sich als ein anderer ausgeben können.
- Unübertragbarkeit: Ein Verifier darf nicht in der Lage sein, sich als ein anderer Teilnehmer auszugeben.“

**2.1.5. Autorisierung**


**2.2. Authentifikationsverfahren – Ein Überblick**

**2.2.1. Wissenbasierte Verfahren**

Diese Klasse von Verfahren basiert auf einem Wissen, das zwischen dem Benutzer und dem Systembetreiber vereinbart wurde. Es wird dabei einerseits unterschieden, ob bei der
2.2. Authentifikationsverfahren – Ein Überblick

Authentifikation das Wissen als solches übertragen wird (und dabei eventuell von einem Angreifer abgefangen und dann für eigene Authentifikationen genutzt werden kann), und andererseits, ob es notwendig ist, das Wissen auf Seiten des Systems (im „Klartext“) abzuspeichern, oder es ausreicht, Ableitungen davon vorzuhalten. Wichtige Vertreter sind:

- passwortbasierte Verfahren,
- Challenge-Response-Verfahren,
- Zero-Knowledge-Verfahren.

Aufgrund der hohen Verbreitung in der Praxis und den damit begründeten Thesen dieser Arbeit liegt im Folgenden der Schwerpunkt bei diesen Verfahren.

2.2.2. Besitzbasierte Verfahren

 Diese Verfahren zeichnen sich dadurch aus, dass der Prover etwas besitzt, zu dem andere Subjekte keinen Zugriff haben. Wichtige Vertreter sind:

- cookiebasierte Verfahren (in Webanwendungen),
- schlüssel- bzw. zertifikatbasierte Verfahren,
- tokenbasierte Verfahren (z. B. Dokumente, Ausweise, Smartcards).

Aufgrund der losen Bindung eines Gegenstands an den Eigentümer werden besitzbasierte häufig mit wissenbasierten Verfahren kombiniert, d. h., das Objekt wird durch eine zusätzliche Wissenskomponente an den Eigentümer gebunden und damit das Objekt vor Fremdnutzung geschützt.

Bewertungskriterien besitzbasierter Verfahren sind insbesondere:

- die Kopierbarkeit,
- die Verfügbarkeit,
- die Interoperabilität

des betrachteten Authentifikationsobjekts.

Beispielsweise hat die Verwendung von Bankkarten mit (kopierbarem) Magnetstreifen gezeigt, dass diese Technologie trotz eines weiteren Authentifikations-Faktors (der PIN) eine Kompromittierung der Identität des Bankkunden durch Skimming-Angriffe begünstigt. In der Folge werden, zumindest in Deutschland, vermehrt Bankkarten mit Chip ausgegeben, es handelt sich somit um (nicht kopierbare) Smartcards.\footnote{Nichtsdestotrotz verfügen solche Bankkarten – aus Kompatibilitätsgründen – in der Regel noch über einen Magnetstreifen, der sich natürlich trotzdem kopieren lässt. Daher fordern Strafverfolger die Abschaffung des Magnetstreifens: \cite{BKA11}.}

Die Verfügbarkeit und die Interoperabilität eines Objekts spielen vor allem in mobilen Szenarien eine wichtige Rolle: Im Gegensatz zu einem Passwort, welches der Prover immer bei sich „trägt“, muss dieser bei besitzbasierten Verfahren daran denken, das notwendige
2. Authentifikationsverfahren

Objekt mitzuführen und – sofern er nicht eigene Geräte benutzt – darauf hoffen, dass Hilfseinrichtungen, wie z. B. ein Kartenlesegerät und dazugehörige Software, überall verfügbar sind. Vor allem auf Smartphones (Mobiltelefone mit erweitertem Funktionsumfang) ist die Interoperabilität und somit die Durchführbarkeit der Authentifikation fraglich, da oftmals diese Hilfseinrichtungen auf PC-Szenarien ausgelegt sind und auf Smartphones nicht funktionieren.


2.2.3. Biometrische Verfahren

Biometrische Verfahren basieren auf der Erfassung und dem Vergleich bekannter Merkmale einer Entität. Die Authentifikation beruht also auf der (zuverlässigen) Bestimmung einer Entität hinter der zu prüfenden Identität. Voraussetzung dafür ist die Existenz von charakteristischen, unveränderlichen Merkmalen für eine Entität. Typische Merkmale, die benutzt werden, sind:

- ein Lichtbild / Passfoto,
- Fingerabdrücke,
- Eigenschaften der Iris einer Person,
- Charakteristika der Sprache,
- Handschrift,
- Hardwareeigenschaften eines Computers (z. B. Seriennummern, Adressen).

Trotz dieser vielversprechenden Techniken sind einige Probleme offenkundig:

1. Es gibt Entitäten, die nicht alle Merkmale in ausreichender Qualität besitzen. Daraus folgt, dass eventuell die Anforderung an die Durchführbarkeit verletzt ist.

2. Die Unveränderlichkeit eines Merkmals stellt nach dessen Kompromittierung ein Problem dar.

3. Die inhärenten Toleranzen dieser Verfahren sperren entweder legitime Benutzer aus oder begünstigen Angriffe auf das System durch illegitime Benutzer.

4. Es ist keine Pseudonymität oder Anonymität möglich.

8Ausreichende Qualität bezieht sich auf die typischen Toleranzen der Erfassungskomponente. So haben beispielsweise manche Personen Fingerabdrücke, die nicht stark genug ausgeprägt sind, um technisch erfasst zu werden.

9Der ehemalige Bundesinnenminister Wolfgang Schäuble dürfte nach der Veröffentlichung von Attrappen seiner Fingerabdrücke durch den Chaos Computer Club Probleme haben, entsprechende Authentifikationssysteme zu benutzen. Siehe [CCC08].
2.3. Passwortbasierte Authentifikationsverfahren


2.3.1. Prinzipielle Probleme und Gegenmaßnahmen

Kopierbarkeit


Erratbarkeit

Ein großes Problem stellt die Wahl des Passworts dar. Wie bei allen kryptographischen Systemen ist es prinzipiell möglich, den Schlüsselraum – also alle Kombinationen theore
10
11

In vielen Anwendungen gibt der Benutzer auch beides zusammen ein und veranlasst die Passwortprüfung durch das System. Dies ist vor allem bei verteilten Systemen sinnvoll, da die Anzahl der Kommunikations- schritte reduziert wird.

Genau genommen muss auch der Loginnname bekannt sein. Dieser ist aber oftmals in (frei zugänglichen) Benutzerverzeichnissen einsehbar oder aus der E-Mail-Adresse des „Opfers“ ableitbar.
2. Authentifikationsverfahren


Um Online-Angriffe abzumildern oder gar zu vermeiden, wird häufig nach einer Anzahl von Fehlversuchen (z. B. nach 3) die nächste Prüfung für eine bestimmte Zeitdauer (Timeout) zurückgewiesen. In manchen Szenarien wird darüber hinaus auch das Benutzerkonto komplett für weitere Versuche gesperrt und muss erst über einen anderen Mechanismus entsperrt werden. Diese Maßnahme ist jedoch genau abzuwägen, da einerseits der Support-Aufwand bei vergessenen Passwörtern steigt und andererseits das Passwortverifikationssystem somit anfällig für Sabotage (Denial-of-Service-Angriffe) wird, vgl. [FS09], Abschnitt 2.2.


Für den Informationsgehalt \( I \) der Zeichen \( z_i \) Alphabets \( Z = z_1, z_2, \ldots, z_n \) gilt

\[
I(z_i) = \log_2 \left( \frac{1}{p_i} \right) = - \log_2(p_i) \quad \text{in Bit pro Zeichen},
\]

wobei \( p_i \) die Wahrscheinlichkeit des Auftretens von \( z_i \) ist. Die Entropie von \( Z \) definiert man wie folgt:

\[
H(Z) = - \sum_{i=1}^{n} p_i \cdot \log_2(p_i).
\]

Wenn man annimmt, dass die Zeichen von \( Z \) gleichverteilte sind, gilt \( p_i = \frac{1}{n} \) für \( i = 1, \ldots, n \).
2.3. Passwortbasierte Authentifikationsverfahren

und somit

\[ H(Z) = - \sum_{i=1}^{n} p_i \cdot \log_2(p_i) = - \sum_{i=1}^{n} \frac{1}{n} \cdot \log_2 \left( \frac{1}{n} \right) = - \log_2 \left( \frac{1}{n} \right) = \log_2(n). \]

(vgl. [Eck08b], Abschnitt 7.4.2).

Beispiel: Ein gleichverteiltes Alphabet bestehend aus Klein- und Großbuchstaben sowie den Ziffern 0 bis 9 (insgesamt 62 Zeichen) entspricht einer Entropie von ca. 6 Bit/Zeichen \((H(Z) = \log_2(62))\). Für ein Passwort, in dem alle Zeichen voneinander stochastisch unabhängig sind, ergibt sich für eine Komplexitätsanforderung von 64 Bit, dass das Passwort mindestens 11 Zeichen lang sein muss.

Verwenden die Benutzer sogar ein Datum als Passwort (dies ist häufig bei sechs- oder achtstelligen Passwörtern der Fall), braucht ein Angreifer nur noch 36500 Möglichkeiten durchzuprobieren, um Erfolg zu haben, wenn man einen Zeitraum von 100 Jahren berücksichtigt (365 Tage · 100), vgl. [FS09], Abschnitt 2.3.

Als Resultat kann man dem Betreiber eines IT-Systems nur anraten, selbst aktuelle Werkzeuge zum Passwortbrechen auf die eigene Benutzerdatenbank anzuwenden und Benutzerkonten mit schwachen Passwörtern nach einer entsprechenden Mitteilung an den Benutzer zu sperren.

Social Engineering


Viele Passwörter und Nutzungshäufigkeit

Mit zunehmender Zahl von Identitäten eines Benutzers treten zwei Probleme auf:


2. Selbst wenn sich Benutzer ersterem Problem bewusst sind, führt dies dazu, dass Passwörter selten genutzter Identitäten vergessen werden und in der Folge Passwort-Rücksetzverfahren nötig sind.

Deshalb haben nahezu alle Webbrowser-Hersteller Möglichkeiten zur Verknüpfung und Hinterlegung von Benutzername-Passwort-Paaren zu Websites realisiert. Eine Site wird im
Allgemeinen anhand des Rechnernamens in der Adresse (URL) eines Dienstes identifiziert. Um einen unbefugten Zugriff auf diese Credentials zu verhindern, bedienen sich die Webbrowser den Zugriffsmechanismen des Betriebssystems („Dateirechte“) und eines (optionalen) *Master-Passworts*.


Der Vorteil dabei ist, dass die Software auf der Seite des Verifiers nicht angepasst werden muss – das Authentifikationsverfahren ist weiterhin passwortbasiert. Diese Browserfunktionen haben auch einen Nachteil: Sie lassen sich nicht ohne die notwendige Software nutzen oder man ist sogar zur Benutzung des betreffenden Kontos an einen Standort bzw. Computer gebunden oder man muss Synchronisierungsssoftware (z. B. Xmarks16) benutzen.

In [Eik11] wird ein Verfahren vorgestellt, welches die Ableitung des Zugangs aus einem Masterpasswort aufgreift, die Berechnung aber dem Benutzer überlässt. Ob die Qualität dieser „generierten“ Passwörter ausreicht, darf bezweifelt werden, da oftmals die Länge des resultierenden Kontopassworts beschränkt und die Site-Kennung nicht als geheim einzustufen ist.


### 2.4. Alternative Authentifikationsverfahren

#### 2.4.1. Einmalpasswörter

Gegen die Bedrohung durch Kompromittierung von Passwörtern hilft eine (zunächst simpel klingende) Maßnahme: Statt immer dasselbe Passwort zu verwenden, wird bei jeder Authentifikation ein neues, nur für *diese* Authentifikation gültiges Passwort benutzt – ein so genanntes Einmalpasswort. Somit kann sich ein Benutzer auch in potentiell unsicheren

---

13 http://www.passwordsitter.de/.
14 http://keepass.info/.
17 Andererseits müssten *alle* Betreiber von Diensten mit Passwort-Authentifikation ihre Anmeldedialoge anpassen.
Umgebungen, beispielsweise Internetcafés, authentifizieren.\(^{18}\) Die Verwendung von Einmalpasswörtern dürfte jedem Online-Banking-Kunden vertraut sein: Zum Abschluss einer Transaktion (z. B. einer Überweisung) ist es nötig, eine Transaktionsnummer (TAN) einzugeben. Die TAN ist ein solches Einmalpasswort. Wie kann ein solches Verfahren umgesetzt werden?

Beim naheliegenden Ansatz vereinbaren Verifier und Prover eine Liste von gültigen Einmalpasswörtern mit optionalem Index (zur Adressierung eines bestimmten Passworts) – analog zur TAN-Liste beim Online-Banking.


**Initialisierung**

Vor der eigentlichen Authentifikation (und nach \(n\) Authentifikationen) muss eine Initialisierungsphase zwischen Verifier und Prover absolviert werden. Dabei berechnet der Prover mit seinem Geheimnis \(g\) und dem Seed \(s\) das Ergebnis der \(n\)-maligen Anwendung einer Hashfunktion \(h\):

\[
\begin{align*}
p_n &= h^{(n)}(g, s) = h(p_{n-1}), \\
p_1 &= h(g, s).
\end{align*}
\]

Das Ergebnis \(p_n\) muss anschließend über einen sicheren Kanal an den Verifier übertragen werden.

**Authentifikation**

Um sich gegenüber dem Verifier zu authentifizieren, berechnet der Prover, nachdem er die Werte für \(k\) und dem Seed vom Verifier erhalten hat, durch \(k\)-malige Anwendung der Hashfunktion den Wert:

\[
p_k = h^{(k)}(g, s).
\]

Der Wert \(k\) entspricht dabei dem Dekrement von \(k\) des vorherigen Authentifikationsdurchlaufs. Um das Einmalpasswort \(p_k\) zu prüfen, braucht der Verifier nur einmal die (vereinbarte) Hashfunktion auf \(p_k\) anzuwenden:

\[
p_{k+1} = h(p_k) = p'_{k+1}.
\]

Stimmt das Ergebnis mit dem gespeicherten Wert \(p'_{k+1}\) (aus dem vorherigen Authentifikationsdurchlauf) überein, ist das Einmalpasswort korrekt und \(p_k\) wird als neuer Referenzwert, \(k-1\) als neuer Index gespeichert.

\(^{18}\) Dabei ist aber zu beachten, dass Einmalpasswörter in der Tat nur gegen Kompromittierung des Passworts schützen. Wird ein Einmalpasswort in einer unsicheren Umgebung zur Authentifikation einer Sitzung benutzt, besteht weiterhin die Gefahr, dass die gesamte Sitzung „gestohlen“ wird. Es sollten daher nur unkritische Anwendungen in solchen Umgebungen eingesetzt werden.
2. Authentifikationsverfahren

Angriffe

Durch aktives Eingreifen in die Kommunikation mittels Man-in-the-Middle-Angriff kann ein Angreifer die Weitergabe des Einmalpassworts \( p_k \) vom Prover an den Verifier unterdrücken und für eine eigene Authentifikation benutzen. Der legitime Prover würde dies am Abbruch der ursprünglichen Transaktion und dem Scheitern einer erneuten Authentifikation mit \( p_k \) merken, da dann \( p_k−1 \) das gültige Passwort wäre.  

Sollte der Prover jedoch im Unklaren über das nächste \( k \) sein und diese Information dem Authentifikationsprotokoll entnehmen, kann der Angreifer gegenüber dem Prover behaupten, \( k−1 \) sei der aktuelle Index, woraufhin der Angreifer \( p_k−1 \) erhielte. Durch Anwendung der Hashfunktion auf \( p_k−1 \) kann der Angreifer \( p_k = h(p_k−1) \) errechnen und mit \( p_k \) eine Transaktion unter der Identität des Provers ausführen. Anschließend würde er die legitime Transaktion mit \( p_k−1 \) weiterleiten. Das Opfer könnte dies nur durch eine verzögerte Ausführungszeit der Transaktion bemerken, was bei Best-Effort-Systemen wie Online-Diensten praktisch nicht erkennbar ist.

Als Maßnahme ist daher vor der Eingabe des Einmalpassworts zwingend der Verifier zu authentifizieren, um diese Art der Angriffe zu verhindern.

2.4.2. Challenge-Response-Verfahren


Nach der Berechnung von \( r_A \) überträgt Alice das Ergebnis zu Bob, der \( r_A \) mit dem Ergebnis seiner Berechnung \( r_B \) vergleicht:

\[
f(s_A, N) = r_A \overset{?}{=} r_B = f(s_B, N).
\]

Sofern Alice nicht die Möglichkeit hatte, den Wert von \( r_B \) zu erraten, ist sichergestellt, dass Alice im Besitz von \( s_A = s_B \) ist.

Angriffe

Eine Bedrohung – unabhängig von der Wahl der Funktion \( f \) – sind Replay-Angriffe. Dabei merkt sich der Angreifer sowohl die Challenges von Bob als auch die (korrekten) Responser von Alice. Wählt nun Bob bei einer späteren Authentifikation die gleiche Challenge, die er zuvor schon einmal an Alice gesendet hat, kann der Angreifer sofort die Response liefern. Es ist jene, die Alice im früheren Durchlauf berechnet hatte. Abhilfe schafft in

\(^{19}\)Bei Einmalpasswörtern haben der Seed und die Iterationszahl die Funktion der Challenge, das Einmalpasswort die Funktion der Response.
2.5. Passwortrücksetzverfahren

diesem Fall z. B. die zusätzliche Verwendung einer monoton steigenden Sequenznummer (beispielsweise die Systemzeit) bei der Bestimmung der Challenge.

Sofern die Übertragung von Challenge und Response über einen unverschlüsselten Kanal erfolgt und für $f$ eine Verschlüsselungsfunktion benutzt wird, kann ein Angreifer prinzipiell einen Known-Plaintext-Angriff durchführen, um den Schlüssel zu ermitteln, mit dem die Response verschlüsselt wurde.


**Wahl der Funktion $f$**

Wie bereits angedeutet beruht die Sicherheit des Verfahrens auf der Annahme, dass der Prover (Alice) bzw. ein Angreifer nicht in Lage ist, die Response zu erraten.

Insofern spielt die Wahl der Funktion eine wichtige Rolle. Wählte man als Funktion beispielsweise die Entschlüsselung einer vom Verifier (Bob) verschlüsselten Challenge, so würde die Response aus dem Klartext (vor der Verschlüsselung beim Verifier) bestehen. Bei ungeschickter Wahl der Challenge oder eines schlechten Zufallszahlengenerators beim Verifier könnte ein Angreifer das Ergebnis unter Umständen voraussagen, ohne tatsächlich die Challenge entschlüsseln zu können (vgl. [BNS05], S. 223).


2.5. Passwortrücksetzverfahren


2. Authentifikationsverfahren


2.5.1. Die PUK – Super-Passwort für dezentrale Entsperrung


2.5.2. Zentrale Maßnahmen

Bei den zentralen Mechanismen besteht die Aufgabe darin, den Systembetreiber davon zu überzeugen, ein neues Passwort für ein angegebenes Benutzerkonto zu setzen. Hierbei ist wiederum eine Form der Authentifikation notwendig.

Persönliches Erscheinen

Die Authentifikation kann beispielsweise dadurch erfolgen, dass der betroffene Benutzer bei der Benutzerbetreuung oder einem Administrator persönlich erscheint und sich mit entsprechenden Ausweisen (z. B. Personal-, Studenten- und/oder Mitarbeiterausweis) als Berechtigter zu erkennen gibt. Der Mitarbeiter trägt anschließend das neue Passwort, aufgrund seiner (größeren) Befugnisse, in das Passwortverwaltungs- system bzw. die Benutzerdatenbank ein. Das Verfahren ist zwar sehr sicher, hat aber erhebliche Nachteile:

1. Es funktioniert nur zu den Geschäfts- bzw. Öffnungszeiten des Supports.
2. Der Benutzer muss u. U. eine räumliche Distanz überwinden, was das Verfahren sehr teuer bis praktisch unmöglich machen kann.

Alternativen

Neben der Rücksetz-Methode „Persönliches Erscheinen“ existieren daher Methoden, die rechnergestützt über das Internet funktionieren. Hierbei initiiert der Benutzer den Rücksetzprozess über eine festgelegte Schnittstelle, in der Regel eine Webapplikation. Zur Authentifikation werden zwei Optionen genutzt:

1. Der Benutzer authentifiziert die Rücksetzanfrage über ein „Super-Passwort“ oder durch die Beantwortung von (vorab festgelegten) Frage-Antwort-Paaren.

16

16 Alternativ erhält der Benutzer eine (einmal gültige) URL zu einer Webapplikation, in der er ein neues Passwort selbst festlegen kann.
2.5. Passwortrücksetzverfahren


3. Die digitale Identität mit dem neuen Personalausweis

Personalausweise erfüllten schon immer eine wichtige Funktion: den Nachweis der Identität des Inhabers durch die ausstellende Behörde. Im Rahmen der Identitätsprüfung einer Person, der Verifizierung, untersucht der Verifier als derjenige, dem ein Ausweis vorgelegt wird, dessen Sicherheitsmerkmale, um die Echtheit des Ausweises festzustellen.

Zum Schutz gegen eine Fremdnutzung erfolgt anhand von Referenzmerkmalen des Inhabers eine Zuordnung von Ausweis zu Inhaber. Klassischerweise wird eine Sichtprüfung des Lichtbilds auf dem Ausweis durchgeführt, zunehmend werden aber auch andere biometrische Merkmale, z. B. Fingerabdrücke, genutzt ([Sch09], S. 7). Stimmen die im Ausweis genannten Merkmalen mit den Merkmalen des Inhabers überein, kann angenommen werden, dass die Identitätsattribute im Ausweis (Name, Anschrift, etc.) dem Ausweisbesitzer entsprechen.


Leider ist neben den Kosten einer Postident-Authentisierung vor allem die sehr hohe Latenz einer Authentisierung, im Vergleich zu anderen E-Commerce-Prozessen, als eines der K.-o.-Kriterien für viele elektronische Anwendungen anzusehen.

1 Streng genommen steht erst nach der Verifikation fest, ob es sich bei demjenigen um den Ausweisinhaber handelt, der den Ausweis vorlegt. Stattdessen wird der Begriff Prover verwendet.
3 Aus diesem Grund erfordert die Prüfung von Berechtigungsnachweisen ohne solche Merkmale, oftmals ebenfalls als „Ausweis“ bezeichnet, dass ein (Identitäts-)Ausweis mit den nötigen Merkmalen hinzugezogen und die Übereinstimmung der Daten auf beiden Dokumenten (Name, ...) geprüft wird.
3. Die digitale Identität mit dem neuen Personalausweis

Andere Anwendungen mit geringeren juristischen Anforderungen greifen deshalb für einen Identitätsnachweis zu Fotokopien von Personalausweisen. Das Problem ist aber, dass dabei die Sicherheitsmerkmale des Ausweises nicht mehr oder nur noch eingeschränkt prüfbar sind, so dass in der Folge der Nachweis nur bedingt gewährleistet ist.

In diesem Kapitel wird zunächst die Technik des neuen Personalausweises (nPA) betrachtet und die Abläufe bei der eID-Funktion beschrieben.

Im zweiten Teil wird untersucht, welche Szenarien und Topologien bei der Umsetzung von eID-Diensten für den neuen Personalausweis prinzipiell möglich sind, und es werden die Vor- und Nachteile dargestellt.

3.1. Der neue Personalausweis

Nachdem seit Oktober 2005 der Reisepass in der Bundesrepublik Deutschland durch eine neue Generation von Dokumenten mit Funkchip (RFID) abgelöst wurde, folgte alsbald auch für den Personalausweis der Beschluss, auf eine neue Version umzusteigen. Seit November 2010 soll der neue, elektronische Personalausweis (kurzform: zunächst ePA, später dann nPA) nicht nur die Fälschungssicherheit für hoheitliche Zwecke verbessern, sondern im Gegensatz zum ePass (dem elektronischen Reisepass) auch weitere Funktionen für privatwirtschaftliche Anwendungen mitbringen (vgl. [BKMN08]).

3.1.1. Authentisierung versus Signatur

In bisherigen Geschäftsvorfällen spielen zwei Prozesse zwischen den Parteien eine besondere Rolle, die beim Entwurf des neuen Personalausweises im Fokus standen: die Authentisierung und die Signaturerstellung, vgl. [BKMN10].


3.1. Der neue Personalausweis

Auch um das Problem der fehlenden Identitätssicherheit zu lösen, haben sich Treuhanddienste etabliert, die als „vertrauenswürdige Dritte“ zwischen beiden Geschäftspartnern vermitteln.\(^5\)

Bei Verträgen in Schriftform reicht eine Authentisierung, die die Identität der Vertragspartner sicherstellt, nicht aus; es bedarf einer expliziten Willensbekundung, dass die Inhalte des Vertrags akzeptiert werden. Durch eine eigenhändige Unterschrift oder eine qualifizierte elektronische Signatur geben die Vertragspartner eine solche auch nachweislich für Dritte (z. B. für einen Richter) ab.

Abbildung 3.1.: Aufbau eines neuen Personalausweises, Quelle: Personalausweisportal

3.1.2. Physische Sicht

Beim neuen Personalausweis (nPA) handelt es sich um eine kontaktlose Smartcard nach [ISO 14443] (RFID), also um eine standardisierte Karte aus Polycarbonat mit eingebettetem Schaltkreis, der drahtlos kommunizieren kann und seine Versorgungsenergie per Induktion aus einem elektromagnetischen Feld bezieht, das vom RFID-Lesegerät erzeugt wird. Die Karte hat die Größe td-1 nach [ICA08] und entspricht von den Abmessungen her somit dem „Scheckkarten-Format“ ID-1.

Vorrangig dient die Karte weiterhin als visueller Ausweis, d. h., dass die bekannten Daten des Ausweisinhabers auch von einem nPA ablesbar und optische Sicherheitsmerkmale vorhanden sind. Zudem ist die von der International Civil Aviation Organization (ICAO) spezifizierte maschinenlesbare Zone (MRZ) auf der Rückseite aufgedruckt, so dass auch der neue Personalausweis als Reisedokument (MRTD) in vielen Ländern genutzt werden kann.\(^6\)

\(^5\)So bietet ein großes Internetauktionshaus an, den Kaufbetrag einer Auktionsware in Empfang zu nehmen und diesen erst dann an den Verkäufer auszuzahlen, wenn die Ware ordnungsgemäß beim Käufer eingetroffen ist.

\(^6\)http://mrtd.icao.int/.
3. Die digitale Identität mit dem neuen Personalausweis

3.1.3. Logische Sicht: Anwendungen

Analog zu anderen etablierten Smartcard-Systemen stellt der nPA mehrere Anwendungen bereit, die über eine Middleware ausgewählt werden können:


- Mit der **Signatur-Anwendung** erhält der Ausweisinhaber die Möglichkeit, die qualifizierte Signatur mit seinem nPA zu nutzen. Voraussetzung dafür ist ein entsprechendes Signaturzertifikat, das über ein Trustcenter bezogen werden muss und nicht vom Aussteller des Ausweises bereit gestellt wird. Der neue Personalausweis ist somit für die Erzeugung qualifizierter Signaturen **vorbereitet**. Diese Anwendung kann auf Wunsch des Inhabers durch Setzen der Signatur-PIN, Erzeugen eines Schlüsselpaares und Installieren eines Signaturzertifikats aktiviert werden, vgl. [TR-03127], S. 34.

Die logische Kommunikation erfolgt nach dem Standard ISO 7816 mittels APDUs.

3.1.4. Zugriff auf die Ausweisdaten

Für den Zugriff auf die hoheitlichen und nicht-hoheitlichen Daten und Funktionen (Altersverifikation, Wohnortabfrage, Pseudonymfunktion) wird ein sogenanntes **Terminal** benötigt. Nach [TR-03127], S. 21ff. gibt es folgende Terminal-Typen:

- hoheitliches Inspektionsystem (zur Polizei- und Grenzkontrolle),
- hoheitliches Authentisierungsterminal (für Änderungen der eID-Daten im Bürgeramt),
- nicht-hoheitliches Authentisierungsterminal (zum Onlinezugriff über das Internet durch Dienstanbieter),
- Signaturterminal (für qualifizierte elektronische Signaturen – QES),
- nicht-authentisieretes Terminal (z. B. die AusweisApp zur Änderung der eID-PIN).
3.2. Die eID-Anwendung

3.1.5. Berechtigungszertifikate

Alle Terminals mit Ausnahme des nicht-authentisierten Terminals, egal ob hoheitlich oder nicht-hoheitlich, benötigen für den Zugriff auf einen nPA ein Berechtigungszertifikat, das von der separaten Public-Key-Infrastruktur mit eigener Zertifizierungsstelle, der CVCA (Country Verifying Certificate Authority, auch „Berechtigungs-CA“ genannt) abgeleitet wurde. Dabei handelt es sich um CV-Zertifikate (Card Verifiable), die für die Prüfung durch Smartcards optimiert wurden. Ein Berechtigungszertifikat beinhaltet unter anderem diese Angaben:

- Referenz auf den Diensteanbieter (CHR – Certificate Holder Reference),
- Referenz auf die DVCA (CAR – Certificate Authority Reference),
- öffentlicher Schlüssel (engl. public key) des Diensteanbieters,
- Zugriffsrechte (CHAT – Certificate Holder Authorisation Template),
- Gültigkeitszeitraum,
- Erweiterungen (vgl. [TR-03127], S. 29f.) wie
  - Angaben zum Diensteanbieter (subjectName, Anschrift, E-Mail-Adresse, Zweck der Anfrage),
  - Name der ausstellenden Berechtigungs-CA (issuerName),
  - Angaben zur zuständigen Datenschutzbehörde,
  - Hashwerte der TLS-Zertifikate des Diensteanbieters (commCertificates).

3.2. Die eID-Anwendung

Mit dem neuen Personalausweis und der eID-Anwendung stellt die Bundesrepublik Deutschland eine sehr fortschrittliche Technologie für die Online-Authentisierung zur Verfügung, die auch hohen Ansprüchen an den Datenschutz gerecht wird.


Neben den aufgedruckten (sichtbaren) Daten eines Ausweisinhabers wie Vorname, Name, Geburtstag und -ort sowie die Anschrift erlaubt der Ausweis – eine entsprechende Berechtigung vorausgesetzt – die Nutzung weiterer Funktionen, die mit dem bisherigen „alten“ Personalausweis nicht möglich waren und aus Sicht des Datenschutz zu begrüßen sind, die:
3. Die digitale Identität mit dem neuen Personalausweis

- Wohnortabfrage, d. h., ein Diensteanbieter kann einen Teilschlüssel gegen die auf dem Ausweis gespeicherte Wohnort-ID – einer leicht modifizierten Variante des Amtlichen Gemeindeschlüssels (AGS) – vergleichen lassen,
- Altersverifikation,
- Pseudonymfunktion (auch Restricted Identification).

So kann ein Online-Dienst beispielsweise prüfen, ob ein Nutzer bereits das 18. Lebensjahr vollendet hat, erhält aber nicht das genaue Geburtsdatum, was eventuelle eine Wiedererkennung des Nutzers ermöglichen würde.

Bei Nutzung der eID-Anwendung eines neuen Personalausweises spielen folgende Entitäten eine zentrale Rolle:
- der Ausweis selbst, bzw. der eingebrachte Chip darin,
- der Ausweisinhaber bzw. der eID-Nutzer,
- der Diensteanbieter – er möchte die Daten auslesen,
- die Zertifizierungsinstanz.

3.2.1. Komponenten

Abbildung 3.2.: Kommunikationsbeziehungen bei der eID-Anwendung (vereinfacht, modifiziert aus [TR-03127]).

Für die Kommunikation mit dem neuen Personalausweis benötigt der Diensteanbieter eine Software-Komponente, die die im Folgenden beschriebenen Schritte im Protokollablauf umsetzt, die notwendigen Daten entsprechend formatiert und ggf. Fehlersituationen verarbeitet und dem Nutzer in geeigneter Form eine Rückmeldung gibt. Diese Komponente wird als eID-Server bezeichnet und stellt das Remote-Terminal dar. Remote-Terminal und
3.2. Die eID-Anwendung

3.2.1. Die eID-Anwendung


Die Kommunikation mit dem Nutzer wird über das Lokale Terminal abgewickelt. Bei Webanwendungen, auf die die eID-Funktion primär abzielt, handelt es sich um verteilte Anwendungen. So läuft ein Teil der Gesamtanwendung auf der Nutzerseite im Webbrowser und der andere Teil auf der Seite des Diensteanbieters. Da sich Personalausweis und Lesegerät auf der Seite des Nutzers befinden, wird eine Teilkomponente benötigt, die einerseits die Kommunikation zwischen Lesegerät/Ausweis und entfernter Anwendung beim Diensteanbieter herstellt und andererseits eine PIN-Eingabe bei Nutzung einfacher Lesegeräte, so genannten Basislesern, ermöglicht.


3.2.2. Ablauf einer eID-Sitzung


Im Anschluss soll der neue (elektronische) Personalausweis vom Diensteanbieter ausgelesen bzw. eine der drei genannten eID-Funktionen genutzt werden, um den Nutzer und somit die in der Sitzung eingegebenen Daten zu authentisieren bzw. eine Berechtigung zu prüfen („Ist der Nutzer volljährig, um auf jugendgefährdende Inhalte zugreifen zu dürfen?“, „Wohnt der Nutzer in Musterstadt?“, „Hat der Nutzer diesen Dienst bereits in der Vergangenheit genutzt?“).


7Die Kommunikation zwischen Lesegerät und Betriebssystem ist durch die PC/SC-Schnittstelle standardisiert.
### 3. Die digitale Identität mit dem neuen Personalausweis

Abbildung 3.3.: Zugriffskontrolle und Authentisierung während einer eID-Sitzung, modifiziert aus [BKMN08].


- Name, Anschrift und E-Mail-Adresse des Diensteanbieters,
- Internetadresse (URL) des Diensteanbieters und der Zweck der Datenübermittlung,
- Hinweis auf die für den Diensteanbieter zuständige Datenschutzbehörde,
- erwünschte Zugriffsrechte (Menge der abzufragenden Berechtigungen),
- Gültigkeitszeitraum des Zertifikates.

Somit ist der erste Teil der gegenseitigen Authentisierung (Identitätsbehauptung des Diensteanbieters) gewährleistet. Nachdem der Nutzer diese Informationen und die auszulesenden Attribute zur Kenntnis genommen hat, bestätigt er seine Einwilligung zum Fortfahren des Ausleseprozesses durch Eingabe seiner eID-PIN.\(^8\) Dabei ist es dem Nutzer gegebenenfalls möglich, einzelne Attribute bzw. Kategorien abzuwählen – diese werden dann vom nPA-Chip nicht an den Diensteanbieter übermittelt. Es ist aber fraglich, ob es in vielen

\(^8\) Der Ausweisinhaber erhält nach Beantragung des nPA eine Transport-PIN zusammen mit einer PUK (zum Zurücksetzen des Fehlbedienungszählers). Die Transport-PIN ermöglicht das Setzen der benutzerdefinierten eID-PIN.
3.2. Die eID-Anwendung

Szenarien möglich sein wird, nur eine Teilmenge der angeforderten Daten zu übermitteln, da eine „Übererhebung“ von personenbezogenen Daten schon wegen des Grundsatzes der Datensparsamkeit auszuschließen ist, oder andernfalls der Diensteanbieter – aufgrund des Fehlens wichtiger Daten – seinen Dienst nicht vollumfänglich anbieten können wird.9


3.2.3. Sicherheit und Datenschutz

Bei der Kommunikation eines Diensteanbieters mit einem nPA im Rahmen der eID-Anwendung sind die Schutzziele Vertraulichkeit, Integrität, Authentizität der Kommunikationspartner, aber auch Anonymität (bzgl. der Wohnortabfrage und Alterverifikation) und der Pseudonymität (bzgl. der Restricted ID) anzustreben. Ein weiteres wichtiges Ziel beim Entwurf des neuen Personalausweises ist die Nutzerorientierung (engl. user centred design): Eine Datenübermittlung erfolgt nur mit Zustimmung des Inhabers. Daneben spielt auch die Möglichkeit einen bestimmten Ausweis jederzeit sperren zu können, eine wichtige Rolle, vgl. [BKMN10], S. 295f.

Für die Bewertung der Sicherheit und der Einhaltung des Datenschutz gilt es die verwendeten Protokolle abzuklopfen. Dabei startet die Analyse bei der Funkzuschnittstelle und geht zu den Protokollen der GAP über.

RFID- und ISO-7816-Schnittstelle


Beim Zugriff auf die logische Smartcard-Sicht (nach ISO 7816) lassen sich nur statische Daten auslesen, die für eine Zuordnung zu einem spezifischen Ausweis und damit zu einem Nutzer nicht ausreichen. Ebenso wurden die Domainparameter für PACE (siehe

3. Die digitale Identität mit dem neuen Personalausweis


PACE

Zur Etablierung eines vertraulichen, integritätsgeschützten und authentisierten Kanals über die per se abhörbare Luft Schnittstelle wurde das PACE-Protokoll [TR-03110] entwickelt. Im Gegensatz zum Protokoll Basic Access Control (BAC), das beim ePass zum Einsatz kommt, erlaubt PACE die Authentisierung und Etablierung kryptographisch starken Schlüsselmaterials auf Basis von kurzen Passwörtern (der 6-stelligen eID-PIN und CAN bzw. einer 5-stelligen Transport-PIN); beim ePass wird hingegen ein 56-Bit langer Access-Key auf Grundlage der maschinenlesbaren Zone (MRZ) benutzt,\(^\text{10}\) vgl. [Eck08a] S. 16. Aus Kompatibilitätsgründen ist zudem die Nutzung der MRZ als Passwort für PACE möglich.

Um ein systematisches Ausprobieren der kurzen PINs und somit einen unzulässigen Zugriff auf den Ausweis-Chip zu verhindern, kommt einerseits ein Fehlbedienungszähler für die eID-PIN zur Anwendung. Andererseits erlaubt weder die (auf dem Ausweis aufgedruckte) CAN noch die MRZ eine Nutzung der eID-Anwendung. Nach zwei Fehlversuchen bei der Verifikation der eID-PIN wird die Anwendung gesperrt, wodurch es möglich ist, einen nPA-Chip (vorübergehend) unbrauchbar zu machen. Ein dritter Versuch lässt sich erst nach einem (erfolgreichen) PACE-Durchlauf mit der CAN als Passwort freischalten. Schlägt auch der dritte Versuch mit der eID-PIN fehl, lässt sich der nPA erst nach korrekter Eingabe des 10-stelligen PIN Unblocking Key (PUK) wieder für die eID nutzen, vgl. [TR-03127], Abschnitt 3.3, S. 16ff.

Terminal Authentication

Im Zuge der Terminal Authentication (TA) erfüllt das Terminal (bzw. Diensteanbieter bei der eID-Anwendung) zwei Aufgaben:

1. die Bereitstellung einer Zertifikatskette, die der nPA-Chip zur Verifikation des Berechtigungszertifikats benötigt,

2. den Nachweis der angekündigten Leseberechtigungen.


\(^{10}\)Die MRZ kann z. B. bei Grenzkontrollen mittels eines optischen Lesegeräts gelesen werden, wofür ein physischer Zugriff auf den Pass bzw. Ausweis notwendig ist.
3.2. Die eID-Anwendung


Passive Authentication


Chip Authentication


Secure Messaging

Die Bindung der auszulesenden Daten an den authentisierten Kanal, aber auch Vertraulichkeit und Integrität der Datenübertragung werden mit den etablierten Verfahren AES für symmetrische Verschlüsselung und Message Authentication Code (MAC) zum Integritätsschutz erreicht.

PKI für Berechtigungszertifikate

Entsprechend der in Abschnitt 3.1.4 genannten Terminal-Typen gibt es für jeden Anwendungsbereich eigene Document Verifier, die die jeweiligen Berechtigungszertifikate ausstellen (vgl. [TR-03127], S. 28):

- Qualitätssicherung beim Ausweishersteller;
- Anwendungen in den Ausweisbehörden – Änderungsdienst/Visualisierung;
3. Die digitale Identität mit dem neuen Personalausweis

- Berechtigungs-CAs für eBusiness/eGovernment-Diensteanbieter;
- hoheitliches Kontrollwesen – Polizei und Grenzkontrolle;

Abbildung 3.4.: Public-Key-Infrastrukturen für den neuen Personalausweises, modifiziert aus [BKMN08].


3.3. eID-Szenarien

Nachdem der potentielle Diensteanbieter mit dem Ablauf einer eID-Sitzung vertraut ist und sich entschieden hat, ein Berechtigungszertifikat zu beantragen, bleibt die Frage: Welche Möglichkeiten zum Betrieb eines eID-Servers gibt es und welche Maßnahmen sind zu ergreifen?

Im Weiteren wird davon ausgegangen, dass die Dienste in Form von Webanwendungen erbracht werden. Andere mögliche Szenarien – wie der oft genannte „Zigarettenautomat“


Bevor die einzelnen Ausprägungen der eID-Server analysiert werden, gilt es, die Schnittstellen eines solchen, aber auch die der Gesamtanwendung („den Dienst“) genauer unter die Lupe zu nehmen. Für eine Bewertung der Szenarien müssen zudem die Anforderungen an einen eID-Einsatz definiert werden.

3.3.1. Komponenten

Im Folgenden wird der Fokus auf der Ebene des Diensteanbieters liegen und die Existenz der Komponente „eID-Server“ vorausgesetzt.

Abbildung 3.5.: Komponenten beim Diensteanbieter (Quelle: [TR-03130]).

Zudem wird es üblicherweise weitere Komponenten geben:

- ein Webserver ist die zentrale Komponente des Diensteanbieters – er bildet die primäre Schnittstelle zum Nutzer, hierüber wird die Webanwendung (die den eigentlichen Dienst darstellt) angesprochen;

- ein Anwendungsserver bildet oftmals die Laufzeitumgebung für die Webanwendung bzw. für Teile davon;

- nahezu obligatorisch ist ein Datenbankserver, der als zentrale Komponente die Daten der Webanwendung und für angeschlossene Prozesse bereitstellt (Prinzip der „Trennung von Form und Inhalt“);
3. Die digitale Identität mit dem neuen Personalausweis

- eine ID-Management-Komponente ist sinnvoll, wenn die Webanwendung über mehrere Authentisierungsquellen verfügt und die eID-Anwendung als weitere Quelle genutzt werden soll (vgl. [TR-03130], Abschnitt 2.4.1);

- die PKI-Komponente deutet den Zugriff auf notwendige Zertifikate, die während der eID-Sitzung benötigt werden (u. a. das Berechtigungszertifikat, CV-Zertifikate und Sperrlisten).

Diese Komponenten können (beispielsweise zur Erhöhung der Verfügbarkeit) auch mehrfach vorgehalten werden.

3.3.2. eID-Server

Schnittstellen

![Abbildung 3.6.: Schnittstellen eines eID-Servers (modifiziert aus [TR-03130]).](image)

Der eID-Server muss eine Reihe von Schnittstellen zur Erfüllung des eID-Service bedienen, Abbildung 3.6 zeigt das Zusammenwirken der Module an den Schnittstellen. Zunächst müssen über eine Management-Schnittstelle bestimmte Parameter, wie Zugangsdaten für angeschlossene Dienste, aber auch Schlüsselmaterial für die Authentisierung der Mandanten, sofern vorhanden, angelegt werden.


Mittels eCard-API kommuniziert der eID-Server mit der Client-Instanz der eCard-API, in der Regel die AusweisApp.
Letztlich beschafft der eID-Server benötigte Zertifikate (von der DVCA und CSCA) und Sperrlisten über die PKI-API von den definierten Diensten, vgl. [TR-03130], Abschnitt 2.2.3.

Aufgaben

Die Komponente eID-Server kapselt zahlreiche Funktionen, unter anderem um diese Aufgaben zu erfüllen, vgl. [VfB11], Abschnitt 2.3:

- **Auslesen der Ausweisdaten und Bereitstellung** für den Dienstanbieter über geeignete Schnittstellen: Dies ist der primäre Grund zum Betrieb des eID-Server.

- Optional eine Mandantenverwaltung. Sofern der eID-Server nicht nur für einen Dienstanbieter agiert, ist es notwendig, Zugangsdaten für die Abholung und Aktualisierung der Berechtigungszertifikate (der Dienstanbieter) abzulegen. Weiterhin muss kryptografisches Schlüsselmaterial für die Kommunikation zwischen Dienstanbieter (Webanwendung) und eID-Server (eID-Schnittstelle) eingerichtet und ggf. aktualisiert werden können.

- **Betrieb eines HSM und Zertifikatspeichers**: Zur sicheren Aufbewahrung von kryptografischen Schlüsseln muss der Betreiber des eID-Servers ein Hardware-Sicherheitsmodul (HSM) bereithalten. Die darin gespeicherten (geheimen) Schlüssel können nicht ausgelesen werden, ein Zugriff erfolgt nur über Funktionen zur Verschlüsselung bzw. Signierung einer zu übergebenen Nachricht. (Typisch ist das Signieren einer kryptografischen Prüfsumme mit einem asymmetrischen Verfahren auf Grundlage eines im HSM gespeicherten privaten Schlüssels.)

  Der Zertifikatspeicher bewahrt das bzw. die Berechtigungszertifikat(e) für den Dienstanbieter, aber auch notwendige CV-Zertifikate der CAs zur Generierung von Zertifikatsketten auf. Ein Zertifikatspeicher erleichtert das Vorhalten (Caching) der notwendigen Zertifikate, die zu aktualisieren sind und somit nicht erst während der eID-Sitzung heruntergeladen werden müssen.

- **Aktualisierung von Berechtigungszertifikaten und Sperrlisten**: Wie bereits dargelegt haben Berechtigungszertifikate für Dienstanbieter nur eine Laufzeit von drei Tagen. Daher ist es erforderlich, Automatismen zu haben, die die Aktualisierung dieses Zertifikats effizient ermöglichen.


3. Die digitale Identität mit dem neuen Personalausweis

3.3.3. Anforderungen an die eID-Infrastruktur


Zur Integration eines eID-Servers in die Infrastruktur des Diensteanbieters gibt das BSI in [TR-03130], Abschnitt 2.3, eine Reihe von Anforderungen vor, die bei Bewertung eines konkreten Szenarios zu berücksichtigen sind.

Da in der Regel personenbezogene Daten zwischen dem Webbrowser und dem Webservver für die Erbringung des Dienstes übertragen werden, ist diese Kommunikation per SSL/TLS (HTTPS) zu sichern. Bei der Ausstellung eines Berechtigungszertifikat wird dafür ein Hashwert des SSL-Zertifikats (X.509) berücksichtigt.

In jedem Fall muss eine Zuordnung der Kanäle zwischen Webbrowser und Webservver sowie zwischen eCard-API-Client (AusweisApp) und eCard-API-Server (eID-Server) vorgenommen werden, um eine missbräuchliche Nutzung zu verhindern. Das notwendige gemeinsame, generierte Geheimnis (PSK) wird über den HTTPS-Kanal zum eCard-API-Client und zum eCard-API-Server übertragen, vgl. [TR-03130], S. 15. Müssen die Daten zwischen Diensteanbieter und eID-Server über offene Netze übertragen werden (entfernter Betrieb des eID-Servers), sind die Daten zu verschlüsseln und zu signieren (ebd.).

Zusätzlich sollten noch diese Anforderungen gestellt werden:


- Es muss (z.B. durch ein Audit) geprüft werden, dass vom eID-Provider keine Authentisierungssprofile erstellt werden (können), vgl. [TR-03130], Abschnitt 2.5: „Die Identitätsdaten werden technisch vom nPA vorgehalten und bereitgestellt und nicht vom eID-Service. Der eID-Service ist daher auch dann kein Identity Provider, wenn er entfernt und ggf. durch andere betrieben wird.“

- Datensparsamkeit: Bei Auslagerung zu externen eID-Providern ist sicherzustellen, dass diese nur Zugriff auf Daten erhalten, die für die Erbringung des eID-Service unbedingt notwendig sind.

3.3.4. Szenario I: Lokaler eID-Server

Im einfachsten Fall wird der eID-Server im Hoheitsbereich des Diensteanbieters betrieben. Dafür muss der Diensteanbieter ein Hardware-Sicherheits-Modul (HSM) gemäß [CPeID], S. 52, vorhalten, in dem die zum Berechtigungszertifikat passenden privaten Schlüssel sicher gespeichert sind. Im Rahmen der Terminal-Authentication wird vom nPA-Chip geprüft, ob der Diensteanbieter Zugriff auf eben diesen Schlüssel hat und im Erfolgsfall Zugriff auf die zugesicherten Daten des Ausweises gewährt. Der Fall, dass der Diensteanbieter
3.3. eID-Szenarien

den eID-Server örtlich getrennt von der Webanwendung betreibt (vgl. [VfB11], Abschnitt 2.4.2), unterscheidet sich technisch wenig von diesem Szenario und wird nicht gesondert betrachtet.

Da für viele Dienstanbieter das Vorhalten dieses HSM und entsprechender Server-Hardware und -Software inkl. Lizenzen unwirtschaftlich ist, besteht eine Möglichkeit darin, diesen „Dienst“ zu einem eID-Provider auszulagern.

3.3.5. Szenario II: Entfernter eID-Server mit Vollzugriff


Hierüber erfolgt nun der Auslesevorgang wie in Abschnitt 3.2.2 erläutert (Anzeige der Anbieter- und Datenschutzinformationen, PIN-Eingabe, PACE, TA, CA, SM).

Sofern das Auslesen der gewünschten Daten erfolgreich war, verpackt der eID-Server diese gemäß der oben beschriebenen WSDL-Definition und signiert das Ergebnis per XML-Signatur mit dem Mandantenspezifischen Signierschlüssel. In Fehlerfällen wird stattdessen eine XML-Nachricht mit der Fehlermeldung an die Webanwendung zurückgegeben.

Da die Webanwendung nicht wissen kann, wann die eID-Anaktion abgeschlossen ist, erfolgt periodisch eine Status-Anfrage (getResultRequest) über die eID-API, bis die Anfrage erfolgreich ist. In Abbildung 3.7 ist der komplette Ablauf einer eID-Anfrage skizziert.

3.3.6. Anwendungstest: eID-Service der Bundesdruckerei

Für den in dieser Arbeit beschriebenen Benutzerverwaltungsdienst wurde der im Rahmen des offenen Anwendungstests bereit gestellte eID-Service der Bundesdruckerei genutzt. Hierbei verhält sich der Ablauf der eID-Kommunikation etwas anders als in Szenario II beschrieben. Für das Auslesen des Personalausweises leitet die Webanwendung beim Dienstanbieter bereits auf Ebene des Anwendungsprotokolls (HTTP) den Webbrowser zu einem Webserver beim eID-Provider (die Bundesdruckerei) weiter und gibt damit die Kontrolle über den Programmablauf der Webanwendung an den eID-Provider ab. Die Webanwendung beim eID-Provider wiederum leitet dann die Initiierung der eCard-API-Sitzung ein, analog zur Beschreibung oben. Der Transport der eID-Daten zwischen Dienstanbieter und eID-Provider erfolgt mittels SAML-Nachrichten über die HTTPS-Verbindungen zwischen Dienstanbieter und Nutzer sowie zwischen Nutzer und eID-Provider. Das Routing entspricht dem SAML-Profil Web Browser SSO Profile sowie dem Redirect/POST-Binding, d. h., die Anfrage vom Dienstanbieter zum Auslesen wird via HTTP-Redirect an den Nutzer und dann an den eID-Provider übertragen, vgl. [OAS08], Abschnitt 5.1.2. Die Antwort des eID-Providers wird über SAML-formatierte Daten per HTTP-POST übermittelt. Das vom
Abbildung 3.7.: Sequenzdiagramm: Funktionaler Ablauf einer eID-Anfrage (Quelle: [TR-03130]).

eID-Provider vorbereitete Formular wird (automatisch) vom Nutzer an eine vordefinierte Adresse der Diensteanwendung (Callback-URL) beim Diensteanbieter abgeschickt.

Für die Authentisierung der Anfrage signiert der Diensteanbieter den SAML-Request mit einem bei der Einrichtung des Mandantenkontos vereinbarten Signierschlüssel. Die Antwort (SAML-Response) wird vom eID-Provider mit einem Mandantenspezifischen Signierschlüssel signiert und mit dem öffentlichen Schlüssel des Mandanten verschlüsselt. So ist sichergestellt, dass außer dem Diensteanbieter niemand die personenbezogenen Daten lesen kann und Manipulationen an den SAML-Nachrichten durch den Diensteanbieter er-
Sicherheitsmaßnahmen

In der Technischen Richtlinie zum eID-Server [TR-03130] wird im Anhang A ebenfalls auf die Eigenschaften eines eID-Server mit SAML-basierter eID-Schnittstelle eingegangen. Die Aussagen beziehen sich dabei auf das Web Browser SSO Profile ([OAS05a], Abschnitt 4.1), das wie erwähnt auch beim eID-Service der Bundesdruckerei benutzt wird.

Es werden die folgenden Maßnahmen vorgeschrieben, um die Kommunikation zwischen eID-Server und Webanwendung beim Diensteanbieter abzusichern:

- **Absicherung der Transportebene mittels SSL/TLS.**
  Dies ist in diesem Szenario durch den geforderten Einsatz von HTTPS als Kommunikationsprotokoll zwischen Benutzer und Diensteanbieter sowie zwischen Benutzer und eID-Provider gewährleistet.

- **XML-Signatur der SAML-Nachrichten.**
  Um die Authentizität der eID-Anfrage und -Antwort zu gewährleisten, werden die entsprechenden SAML-Nachrichten signiert. Die Signatur erfolgt über das vollständige XML-Dokument, das dem SAML-Protokoll zugrunde liegt.

- **XML-Verschlüsselung in den SAML-Nachrichten.**
3. Die digitale Identität mit dem neuen Personalausweis

- **Bindung der drei Kommunikationskanäle einer Sitzung aneinander.**
  Um die Gültigkeit der im Zuge der eID-Sitzung ausgetauschten SAML-Nachrichten auf genau diese Sitzung zu beschränken, ist es erforderlich die drei Kanäle
  - Benutzer ↔ Diensteanbieter,
  - Benutzer ↔ Webserver beim eID-Provider und
  - eCard-API-Client ↔ eCard-API-Server

miteinander zu verknüpfen. Hierfür wird jeweils ein gemeinsames Geheimnis (engl. preshared key – PSK) verwendet, so dass sich kein Dritter per Man-in-the-Middle-Angriff in die Kommunikation einklinken kann.

- **Verwendung von Zeitstempeln in den SAML-Nachrichten.**

Replay-Angriffe funktionieren somit nur innerhalb dieses kurzen Zeitfensters. Für Denial-of-Service-Angriffe müsste der Angreifer – sofern sich der Dienst die Kombination (Zeitstempel, Signaturersteller) merkt – jeweils „frische“, korrekt signierte SAML-Nachrichten erzeugen, was einen solchen Angriff aber zu teuer werden lässt.


Wie in [OAS05b] neben den soeben genannten Gefährdungen beschrieben, muss sich der Betreiber des Dienstes, aber auch der des eID-Servers im Klaren sein, dass die Sicherheit der SAML-Kommunikation und der aufgestellten Identitätsbehauptungen von neuralgischen Punkten abhängen. Zum einen kann eine SAML- Assertion offensichtlich nur so aussagekräftig sein wie das Resultat ihrer Erfassung. Die Assertion hängt also unmittelbar von der eID-Anwendung auf der Seite des eID-Providers ab. Zum anderen ist ebenfalls ein Augenmerk auf die Anforderungen der benutzen Sicherheitsprotokolle zu richten. So sind die kryptographischen Schlüssel über einen **sicheren** Kanal auszutauschen und sicher bei den beteiligten Parteien zu speichern. Schließlich ist für die Durchführung der Protokolle ein
kryptographisch sicherer Zufallszahlengenerator erforderlich, da sonst eventuell benutzte Ephemeralschlüssel vorhersagbar wären.

Als Anmerkung wird in [TR-03130], Anhang A 2.2, von der Verwendung des Redirect-Bindings abgeraten, wie es für die Übermittlung des SAML-Request während des Anwendungstests eingesetzt wurde.\(^\text{13}\) Da dabei die SAML-Daten und die Signatur in der URL kodiert werden und die maximale Größe der URL von der Implementierung des Webbrowsers abhängt, kann nicht sicher gestellt werden, dass die Übermittlung des SAML-Request zu Fehlern führt. Andererseits muss dabei auch berücksichtigt werden, dass für das Nutzungszenario nur sehr wenige Webbrowser in Frage kommen, für die ein AusweisApp-Plugin bereit steht. Alle Browser, die im Rahmen der Implementierung zusammen mit dem eID-Service der Bundesdruckerei getestet wurden, zeigten hier keine Anfälligkeiten – die Größe des SAML-Requests ist zudem beschränkt, was eingabeabhängige Laufzeitfehler ausschließt. Dennoch sollte das Problem bei der Portierung auf andere Geräte (beispielsweise Smartphones) im Zusammenhang mit SAML-basierten eID-Servers berücksichtigt werden.

### 3.3.7. Szenario III: Entfernter eID-Server mit eingeschränktem Zugriff

Das in Abschnitt 3.3.5 beschriebene Szenario hat ein inhärentes Problem: Es laufen sehr viele personenbezogene Daten bei wenigen eID-Providern an wenigen zentralen Stellen auf, so dass prinzipiell Bewegungsprofile der Form „Nutzer \(A\) authentisiert sich beim Online-Shop \(B\)“ oder auch „Nutzer \(C\) fragt Daten von Behörde \(D\) ab“ erstellt werden könnten. Aufgrund von Datenschutzbestimmungen und Zertifizierungen soll dies zwar unterbunden werden, dennoch wecken solche „Daten-Knotenpunkte“ Begehrlichkeiten.


\(^{13}\)In neueren Versionen der Testinfrastruktur der Bundesdruckerei wird hingegen, wie in der [TR-03130] empfohlen, nur noch das POST-Binding benutzt und die Daten im SAML-Request sind zudem für den eID-Server verschlüsselt.
vom eCard-API-Client zurückgeliefert werden, zur weiteren Verarbeitung bzw. Bereitstellung für die Webanwendung. Somit würde sichergestellt, dass nur der (dezentrale) Diensteanbieter die für ihn bestimmten Daten „zu Gesicht bekommt“.

3.3.8. Fazit

Die vorgestellten Szenarien und Abläufe zeigen, dass die eID-Schnittstellen und die eCard-API flexibel genug sind, den Betrieb eines notwendigen eID-Servers an spezialisierte eID-Provider auszulagern.

Vor allem der Betrieb vorgeschriebener Komponenten, wie eines HSM, stellt für viele Diensteanbieter (z.B. Online-Shops) eine Hürde dar, wenn die Anwendung in einem Rechenzentrum betrieben wird. Dort ist in der Regel die Installation von kundenspezifischer Hardware nicht möglich.


Bei der Spezifikation des eID-Servers hat das BSI aber leider auch die Chance vertan, zentrale Stellen zu vermeiden, an denen Daten ausgelesen und prinzipiell technisch nachverfolgt werden können, auch wenn dies durch Policies verboten ist.
4. Entwurf eines Benutzerverwaltungsdienstes

Nachdem im vorherigen Kapitel der neue Personalausweis und eID-Infrastrukturen vorgestellt wurden, wird nun der Entwurf eines Benutzerverwaltungsdienstes, der auf diesen Komponenten aufbaut, beschrieben.

4.1. Zielstellung


In dieser Arbeit wird ein Dienst vorgestellt, der die Pseudonymfunktion des neuen Personalausweises als starken Zwei-Faktor-Authentifikator benutzt. Auf dieser Grundlage kann der Benutzer ein entscheidendes Attribut des eigenen Accounts verwalten – das Passwort für den alltäglichen (Rechner-)Login.

Zudem soll untersucht werden, ob sogar neue Benutzerkonten mit Hilfe des Dienstes erzeugt werden können.

Dafür soll ein Benutzerverwaltungsdienst prototypisch für das Fallbeispiel Benutzerkontenverwaltung am Institut für Informatik der Humboldt-Universität zu Berlin (HUB) entwickelt werden, der die prinzipielle Machbarkeit darlegen und Probleme verdeutlichen soll. Auf die Implementierungsdetails wird im anschließenden Kapitel eingegangen.

4.2. Fragestellungen

- Welche Voraussetzungen müssen erfüllt sein, um die komplette Verwaltung von Benutzerkonten auf Basis des neuen Personalausweises zu ermöglichen?
  Die komplette Verwaltung umfasst die Erzeugung eines Accounts sowie das Ändern, Verlängern und Löschen bestehender Accounts.

- Wie kann ein neuer Personalausweis mit einem bestehenden Account verknüpft werden, wenn der bislang verknüpfte Ausweis verloren wird oder abgelaufen ist?

- Welche Maßnahmen sind zu ergreifen, um die zu speichernden personenbezogenen Daten vor Missbrauch zu schützen?
4. Entwurf eines Benutzerverwaltungsdienstes

4.3. Identitätsmanagement

Der Begriff Identitätsmanagement (IdM – engl. identity management), teilweise auch Identity and Access Management (IAM), bezeichnet eine Reihe von Maßnahmen die notwendig sind, um die Benutzer eines IT-System zu identifizieren und sicherzustellen, dass diese nur unbedingt erforderliche Zugriffe auf das System und seine Ressourcen erhalten, vgl. [MA07], S. 10.

Im einfachsten Fall sind dies Maßnahmen zum

- Anlegen,
- Ändern und
- Löschen


Abbildung 4.1.: Ein Ebenen-Modell für das Identitätsmanagement (Quelle: [MA07])

Zur Veranschaulichung soll das Modell anhand des Ist-Zustands der Benutzerverwaltung am Institut für Informatik der Humboldt-Universität zu Berlin erläutert werden.

4.3.1. Ist-Zustand

Die Personendaten-Ebene bildet die Basis aller Ebenen. Hier werden Daten der Personen gesammelt und für die IT-Systeme der Organisation bereit gestellt. In Firmen werden die
4.3. Identitätsmanagement


In der Autorisierungs-Ebene werden Berechtigungen für den Zugriff auf die Ressourcen (in Ebene 2) erteilt. Abstrakt kann der Administrator dafür ein Rollen-Modell benutzen. „Eine Rolle ist eine Sammlung von Berechtigungen, die notwendig sind, um eine bestimmte Aufgabe zu erfüllen“ ([\text{MA07}], S. 16). Entscheidend ist dabei, dass ein Benutzer immer nur eine Rolle zur Erfüllung der Aufgabe erhält und bei einer neuen Aufgabe die alte Rolle und somit die Berechtigungen abgibt, um eine neue Rolle zu erhalten. Praktisch kann dies

\(^1\)Die Rechnerbetriebsgruppe (RBG) sorgt – wie der Name es andeutet – für den reibungslosen Betrieb des Rechnerbetriebs am Institut. Analog zu einem „kleinen Rechenzentrum“ stellt sie Dienste für die Studenten und Mitarbeiter bereit und koordiniert die Anbindung von IT-Systemen an den Lehrstühlen.

\(^2\)Eine E-Mail an die „Alias-Adresse“ des Benutzers wird an dasselbe Postfach des Benutzers zugestellt wie bei einer E-Mail an die primäre E-Mail-Adresse.
4. Entwurf eines Benutzerverwaltungsdienstes


4.3.2. Speicherung der Personendaten


Verzeichnisdienste stellen strukturierte Daten an zentraler Stelle in einer Organisation über ein Netzwerk zur Verfügung. Das zugrunde liegende Verzeichnis kann im einfachsten Fall als Liste realisiert sein, z. B. ein Telefon-Verzeichnis bzw. Telefonbuch. Hierarchische Verzeichnisse hingegen erlauben aufgrund ihrer Baumstruktur, die als Directory Information Tree (DIT) bezeichnet wird, die Abbildung von Beziehungen und Abhängigkeiten.


- bind/unbind – Verbindungsaufbau und -abbau zu einem LDAP-Server (inkl. Authentifizierung);
- search – Suchen nach einem Objekt gemäß eines angegebenen Suchfilters;

3https://goya3.informatik.hu-berlin.de/goyacs.
4.3. Identitätsmanagement

- **add/delete/modify** – Hinzufügen, Löschen und Ändern der Attribute eines Objekts;
- **compare** – Vergleich eines Attribut-/Wert-Paares mit den korrespondierenden Attributen des im DIT gespeicherten Objekts (DN);
- **abandon** – Abbruch einer Anfrage, die zuvor ausgeführt, aber noch nicht beendet wurde;
- **modify RDN** – ermöglicht Verschieben von Objekten (Knoten) und Teilbäumen eines DIT.

In einem Verzeichnis nach X.500 können die Objekte über einen absoluten Namen, den *Distinguished Name* (DN), oder über den (zum Elternobjekt) relativen Namen *Relative Distinguished Name* (RDN) adressiert werden. Die Daten im Verzeichnis werden auf Basis sogenannter Schemata strukturiert. Ein Schema definiert unter anderem die Datentypen von Attributen und die Verkettung von Attributtypen in Objektklassen. Analog zur objektorientierten Programmierung besitzen alle Instanzen einer solchen Klasse die definierten Attribute, die verpflichtend oder optional sein können, und es können Klassen im Zuge der Vererbung voneinander abgeleitet sein. (Zum Beispiel ist der Nachname einer Person zwingend, die Telefonnummer hingegen nicht.)


Im Gegensatz dazu sind Daten in einem relationalen Datenbanksystem sehr viel einfacher strukturiert. Die Objekte der Relation (auch Datensätze genannt) sind Tupel, die aus einer Kombination von Attributwerten (die Attributnamen zugeordnet sind) bestehen. Anschaulich beschreibt eine Relation somit eine Tabelle in der Datenbank. Attribute sind in der Regel typisiert (Integer, String, Boolean, etc.). Diese einfache Form zusammen mit einer Abfragesprache, wie der Structured Query Language (SQL), die es ermöglicht, Daten flexibel zu verknüpfen (über die Operationen SELECT, JOIN, UNION, Subqueries und Aggregationen), macht Relationale Datenbanksysteme attraktiv für Anwendungen, in denen große Mengen gleichartiger Daten anfallen. So werden diese Systeme gern dann zur Personen- und Account-Datenverwaltung eingesetzt, wenn das Datenbanksystem sowieso bereits vorhanden ist und nur wenige Anwendungen auf den Datenbestand auf-
4. Entwurf eines Benutzerverwaltungsdienstes

setzen. LDAP-Verzeichnisse zielen vor allem auf solche Szenarien ab, in denen viele unterschiedliche Systeme Zugriff auf Benutzerdaten benötigen. Dies lässt sich beispielsweise daran ablesen, dass fast alle Produkte, die eine Authentifizierung durchführen, eine LDAP-Schnittstelle bereitstellen.

Diese Arbeit beschränkt sich im Weiteren auf die Betrachtung der Verwaltung von Personendaten und darauf aufbauend den Benutzerkonten (Ebene 1 und 2 im vorgestellten Modell). Insbesondere wird nicht auf die Verwaltung von Zugriffsrechten eingegangen. Im Prototyp wird die Funktion des Identitätsmanagements durch die betrachtete Webanwendung umgesetzt.


4.4. Prozesse

Aus der Zielstellung ergeben sich unmittelbar drei Prozesse, die der Dienst zur Realisierung umsetzen muss. Diese sollen im Folgenden umrissen werden.

4.4.1. Zurücksetzen des Account-Passworts

Die zentrale Funktion des Dienstes, die von der eID-Anwendung des neuen Personalausweises am meisten profitiert und einen echten Mehrwert für den Benutzer bringt, ist das Zurücksetzen des Account-Passworts.\(^5\)


\(^5\)Die (automatische) Erfassung der personenbezogenen Daten aus einem Personalausweis ist aus organisatorischen und Sicherheitsgründen zu begrüßen, würde aber mehr dem Rechnerbetrieb denn dem Benutzer helfen.

Diese Funktion setzt voraus, dass zuvor bereits eine Zuordnung von nPA zu Account durchgeführt wurde.

### 4.4.2. Verknüpfen eines bestehenden Accounts mit einem neuen Ausweis


Die Forderung nach einem starken Zwei-Faktor-Authentifikator – der gerade ein Vorteil gegenüber anderen Lösungen, wie die in Abschnitt 4.3.2 beschriebene PIN-Nutzung beim CMS, ist – führt an dieser Stelle zu einem Problem: Wie kann mit den zur Verfügung stehenden Mitteln (gültiges Passwort, PIN der Studienbuchseite, etc.) die rID des neuen Ausweises hinterlegt werden, ohne damit das Sicherheitsniveau des nPA als Authentifikator zu gefährden und auf das Niveau der anderen Mittel zu reduzieren.


Alternativ könnte nach einem Login mit dem Account-Passwort als Transaktionslegitimation ein Einmalpasswort bzw. eine TAN zur Autorisierung des Benutzers für diese Funktion des Dienstes zum Einsatz kommen. Auf diese Art würde eine Zwei-Faktor-Authentifizierung (Wissen und Wissen) benutzt, die durch das Einmalpasswort ein fast so

---

6Hierfür könnte der Angreifer theoretisch sogar seinen eigenen nPA benutzen. Da nur die Restricted ID vom Dienst gelesen wird, wäre die Identität des Angreifers hinter dem Pseudonym „geschützt“.
4. Entwurf eines Benutzerverwaltungsdienstes


4.4.3. Anlegen eines neuen Accounts


Durch den Dienst muss zudem sichergestellt werden, dass nach einer Zuweisung nPA → Benutzerkonto keine weitere Zuweisung möglich ist, um einen Missbrauch durch Anlegen beliebig vieler Accounts zu verhindern.


Grundsätzlich besteht für die Realisierung dieser Funktion ein Problem: Wie kann sichergestellt werden, dass die Zuordnung nPA → Neues Benutzerkonto nur für Angehörige der Organisation möglich ist? Hierfür stehen unter anderem folgende Maßnahmen zur Verfügung:

1. Die Beschränkung der Verfügbarkeit des Dienstes auf einen Standort bzw. das lokale Netzwerk. Diese Variante empfiehlt sich tendenziell für Firmen, da hier die neuen Mitarbeiter in der Regel vor Ort sind. Im universitären Umfeld könnte dies jedoch den

---

7Gemäß Benutzungsordnung des Instituts für Informatik der HU Berlin, §3, Absatz 6.
8Das Institut bzw. die Universität tritt als Internetdienstanbieter (engl. internet service provider, ISP) auf und muss gemäß §§ 113 TKG, Abs. 1 in Verbindung mit §§ 111 TKG den Namen und die Anschrift des Benutzers sowie das „Datum des Vertragsbeginns“ erheben, speichern und den im Gesetz genannten „zuständigen Stellen auf deren Verlangen unverzüglich Auskünfte über die […] erhobenen Daten […] erteilen.“
4.5. Entwurfsentscheidungen

erwünschten Effekt (Entlastung der Benutzerbetreuung) verhindern, da neue Studen-
ten solche administrativen Aufgaben noch vor Studienbeginn erledigen werden und
dafür ein Zugriff von zu Hause wünschenswert ist.

2. Die Ausweisdaten werden mit „Stammdaten“ abgeglichen. Existiert eine Person im
Repository ohne Benutzerkonto und stimmen die Daten aus der Immatrikulation
bei Studenten bzw. der Einstellung bei Mitarbeitern (Name, Vorname, Geburtsda-
tum, Anschrift) mit den Ausweisdaten überein, wird der Account angelegt und mit
dem nPA verknüpft. Dazu bedarf es aber intelligenter Matching-Methoden, wie dem
Fuzzy-Matching oder einer Menge von regulären Ausdrücken, um eine Übereinstim-
mung auch bei kleinen Abweichungen zu ermöglichen.

3. Die Wiedererkennung eines Angehörigen mittels eines zuvor vereinbarten Geheim-
nisses. So könnte, wie in den vorgestellten Anwendungen beim Informatik-Institut
und CMS, eine Zuordnung anhand der Matrikelnummer oder PIN erfolgen.

4. Ableitung eines neuen nPA-basierten Accounts von einem bestehenden Account ei-
er anderen Abteilung der Organisation. Beispielsweise kann ein neuer Informatik-
Account auch nach Verifizierung der Credentials eines CMS-Accounts angelegt wer-
den.\(^\text{10}\) Die Bindung des Benutzers an die Organisation ist in dem Fall schon bei der
anderen Abteilung erfolgt.

5. Alternativ könnte bei einer elektronischen Immatrikulation auf Basis des nPA die
Restricted ID des Studenten in der Studenten-Datenbank hinterlegt werden:
Ein elektronisches (Web-)Formular wird mit den personenbezogenen Daten der eID-
Anwendung ausgefüllt (Name, Vorname, Anschrift, Geburtsdatum)\(^\text{11}\), und zusätzli-
che Daten können durch den künftigen Studenten ergänzt werden. Im Anschluss
druckt dieser eine Bestätigung aus, welche eine eindeutige Nummer zur späteren
Zuordnung enthält, und schickt diese unterschrieben mit den notwendigen Anlagen
zum Immatrikulationsbüro. Einrichtungsspezifische Accounts könnten dann auf Ba-
sis dieser rID/Stammdaten angelegt werden.

4.5. Entwurfsentscheidungen

Um aus den funktionellen Anforderungen, die wie dargestellt noch Optionen anbieten,
einen Prototyp erstellen zu können, werden folgende Entscheidungen für den Entwurf der
Architektur und die spätere Implementierung getroffen:

- Da der eID-Service der Bundesdruckerei zusammen mit der AusweisApp für den
Zugriff auf einen nPA genutzt werden soll, wird der Benutzerverwaltungsdienst als
Webanwendung umgesetzt. Die eID-Kommunikation erfolgt wie im eID-Szenario II
vorgestellt, siehe auch: Abschnitt 3.3.5.

\(^\text{10}\) Diese Variante wird zur Vereinfachung der Prozesse zukünftig ausschließlich zur Erzeugung von Informatik-
Accounts verwendet werden.

\(^\text{11}\) Diese Daten werden bei einer Immatrikulation generell erhoben, so dass keine datenschutzrechtlichen Be-
denken zu erwarten sind.
4. Entwurf eines Benutzerverwaltungsdienstes

- Trotz dieser Festlegung bzgl. des eID-Providers soll die Tatsache berücksichtigt werden, dass der Betreiber später eventuell einen anderen eID-Provider nutzen möchte, der sich in der eID-Schnittstelle von jener der Bundesdruckerei unterscheidet. Das heißt, die Software soll prinzipiell ohne große Anpassungen unterschiedliche eID-Server anbinden können.


- Zur Vereinfachung der Implementierung kann nur ein Benutzerkonto (das „primäre“) mit dem nPA verknüpft werden. Andere Implementierungen könnten über die Abfrage des Loginnamens mehrere Accounts einem nPA zuordnen. In der Praxis dürfte diese Anforderung aber nur für wenige Benutzer, oftmals Angehörige des IT-Supports (zur Realisierung unterschiedlicher Rollen), relevant sein.


4.5.1. Umsetzung der Funktionen im Prototyp

Aus den gezeigten Alternativen werden für den Prototyp folgende Festlegungen getroffen:

1. Für die Erstellung eines Accounts wird eine gültige, nicht zugeordnete Matrikelnummer abgefragt und geprüft, ob diese auf einer Referenzliste und zudem nicht auf einer

---

12Diese Technik wird als Proxy bzw. ReverseProxy bezeichnet.

Bei Unstimmigkeiten (Matrikelnummer nicht gültig, Passwort unterschiedlich, etc.), ebenso wie bei einer bereits registrierten Restricted ID erhält der Benutzer eine Fehlermeldung.


Ist die rID schon einmal (auf diesen oder einen anderen Account) zugeordnet worden, erhält der Benutzer eine Fehlermeldung, siehe Screenshot A.10 im Anhang.

4.6. Architektur

Für die Komposition des Dienstes werden eine Reihe von Subsystemen benötigt. Ein Teil der Systeme wurde bereits bei den eID-Szenarien in Abschnitt 3.3 beschrieben.

Abbildung 4.3.: Komponenten des Benutzerverwaltungsdienstes
4. Entwurf eines Benutzerverwaltungsdienstes

4.6.1. Webserver


4.6.2. Anwendungsserver

Der Anwendungsserver (oder Applikationsserver) führt die Dienstanwendung aus und stellt dafür die Ausführungsumgebung zur Verfügung, z. B. benötigte Interpreter und Software-Bibliotheken, aber auch ein Modul, das per HTTP mit der „Außenwelt“ kommuniziert. Er verbindet sich ferner mit dem eID-Service, um die Ausweisdaten des Benutzers zu erhalten und auf dieser Grundlage den weiteren Ablauf einer Sitzung der Webanwendung zu steuern. Hierfür werden zwei Einsprung-Adressen für die Anwendung eingerichtet:


Der Systemadministrator konfiguriert auf diesem Server die notwendigen Schlüssel (Verschlüsselung und Signatur) für die Kommunikation mit dem eID-Server.

Es empfiehlt sich diese Komponente aufgrund der hohen Komplexität weiter zu verfeinern. Eine sinnvolle Modularisierung wird im Abschnitt 4.6.6 präsentiert.

4.6.3. Verzeichnisdienst

Wie in Abschnitt 4.3.2 geschildert, kommt am Institut für Informatik ein LDAP-basierter Verzeichnisdienst zum Einsatz. Dieser soll auch beim Dienst als Repository für die Benutzerdaten und Credentials dienen. Darüber hinaus gibt es zwei Möglichkeiten, wie die für die beschriebenen Prozesse notwendige Restricted ID im Verzeichnis abgelegt werden kann. Im einfachsten Fall benutzt man ein bislang nicht verwendetes Feld der Objektklasse des Standard-Schemas. Nahezu alle LDAP-Implementierungen verwenden zur Abbildung von Personendaten die Objektklassen person und darauf aufbauend inetOrgPerson. Bei letzterer böte sich scheinbar das Attribut jpegPhoto an, welches generell Binärdaten wie die Restricted ID unterstützt. Ungünstigerweise lässt sich nach diesem Attribut nicht suchen, was aber für den behandelten Dienst notwendig ist. Ein anderes Feld, das eine Suche ermöglicht und – zumindest am Institut für Informatik – nicht benutzt wird, ist das Feld description. Da es sich hierbei um ein Attribut für Zeichenketten handelt, muss die Restricted ID mittels Base64 kodiert bzw. dekodiert werden. Da das Attribut die EQUALITY „caseIgnoreMatch“

---

13Base64 beschreibt eine Möglichkeit, beliebige Binärdaten (Octets) mit druckbaren Zeichen darzustellen. Dabei werden jeweils drei ursprüngliche Zeichen (zu 8-Bit) in vier druckbare Zeichen (zu 6 Bit), also insgesamt 24 Bit, umgewandelt. Der Name entspricht der Zahlenbasis des Zeichenvorrats von \(2^6 = 64\) Zeichen. Als Zeichen des Base64-Alphabets werden die großen und kleinen Zeichen des lateinischen Alphabets (52 Zeichen) zzgl. der Ziffern 0 bis 9 sowie „+“ und „/“ benutzt. Das Gleichheitszeichen „=“ dient zum Auffüllen (Padding), wenn gilt: Länge(Binärdaten) mod 3 \(\neq 0\).
besitzt, ist zudem das Ergebnis durch die Anwendung mit dem Suchfilter zu vergleichen, auch wenn eine Übereinstimmung trotz nicht berücksichtigter Groß-/Kleinschreibung des Base64-Strings unwahrscheinlich erscheinen mag.

Der empfohlene Weg zur Speicherung der Restricted ID ist jedoch das Installieren eines passenden Schemas für die Restricted ID. Alle anderen eID-Daten haben bereits eine Entsprechung in den Standard-Schemata.

### 4.6.4. eID-Server


### 4.6.5. Testlogin-System


Im operativen Betrieb wird man in der Regel eine Prüfanwendung statt eines separaten Servers benutzen, der die äquivalenten Abfragen beim LDAP-Server durchführt und das Ergebnis in geeigneter Weise aufbereitet.

### 4.6.6. Anwendungsmodul

Für die Verfeinerung der Anwendung werden drei Arten von Modulen benutzt: Frontend-Module bilden die besprochenen Funktionen auf eine Benutzerschnittstelle ab und steuern den Aufruf der für die Realisierung notwendigen Methoden aus den anderen Modultypen. Über eigene URLs lassen sich die jeweiligen Frontendmodule und dementsprechend die einzelnen Funktionen direkt adressieren. Somit kann die Funktionalität des Dien-
Entwurf eines Benutzerverwaltungsdienstes

4. Entwurf eines Benutzerverwaltungsdienstes


**Frontend-Module**


**Backend-Module**


**eID-Modul**

Gemäß der benutzten eID-Schnittstelle der Bundesdruckerei über SAML-Nachrichten setzt das eID-Modul (für die Bundesdruckerei) die beiden folgenden Teile der Kommunikation um:

1. Die Erzeugung des (signierten und verschlüsselten) SAML-Requests mit den vom Frontend-Modul geforderten Datengruppen und Funktionen. (z. B. „Lese Vorname, Nachname, Anschrift und Restricted ID aus dem Ausweis des Benutzers!”)

2. Die Verarbeitung der SAML-Response vom eID-Server. Dazu muss die Nachricht entschlüsselt und die Signatur (vom eID-Provider) geprüft werden. Anschließend sind die personenbezogenen Daten in geeigneter Form (beispielsweise als Instanz der Klasse „Person“) der Callback-Methode des Frontendmoduls zu übergeben.

---

14Der Standard IEEE 802.1X erlaubt eine Authentifizierung in Ethernet-Netzwerken, insbesondere die eines WLAN-Clients (Station) unter Nutzung eines X.509-Zertifikats. Der Nachweis über den Besitz des zugehörigen privaten Schlüssels wird mittels Challenge-Response erbracht.
4.7. Datenschutzrechtliche Betrachtungen

Bei der Untersuchung der Fragestellung bzgl. des Datenschutzes der personenbezogenen Daten sollen drei Aspekte erörtert werden:

1. Welche Daten werden von einem Benutzer erhoben und gespeichert?
2. Aus welchem Grund werden die Daten erhoben?
3. Wer kann auf welche Daten zugreifen?

4.7.1. Betroffene Datengruppen

Grundsätzlich muss man bei der Betrachtung, welche Daten vom vorgestellten Dienst bearbeitet werden, zwischen Stammdaten und Bewegungsdaten unterscheiden.

Stammdaten sind jene, die einen Benutzer unmittelbar von anderen unterscheidbar machen und die sich nur selten bis gar nicht ändern. Im Gegensatz dazu können Bewegungsdaten zunächst einmal nur einer Gruppe von Personen zugeordnet werden und sind erst bei Verknüpfung mit anderen (Stamm-)Daten einem Individuum zuzuordnen (Beispiel: Accountname → Name und Anschrift).

Dennoch bergen Bewegungsdaten eine gewisse Brisanz in sich, da sie Rückschlüsse auf Präferenzen und Eigenschaften von Benutzern zulassen können, die die Intimsphäre der Personen betreffen und somit rechtlich besonders geschützt sind.

Zu den vom Dienst betroffenen Stammdaten gehören:

- Nachname des Benutzers,
- Vorname des Benutzers,
- Anschrift des Benutzers,
- Restricted ID (rID) bzgl. des Dienstes und des Personalausweises des Benutzers.

Auf folgende Bewegungsdaten kann der Dienst zugreifen:

- IP-Adresse des Rechners bzw. Anschlusses der Benutzers (Zugriff auf die Webanwendung),
- Daten, die im Zuge eines Auditings der Anwendung erhoben werden.

Außerhalb des Zugriffs des Diensteanbieters fallen zudem Bewegungsdaten der eID-Sitzung beim eID-Provider an. Diese werden in dieser Arbeit nicht weiter betrachtet. Da das Auditing als optionales Modul angesehen wird, kann durch den Dienst der Umfang der Datenerhebung nicht abgeschätzt werden. (Prinzipiell können natürlich alle erhobenen eID-Daten und Nutzereingaben sowie Netzwerkdaten etc. betroffen sein.)
4.7.2. Datenerhebung und -speicherung


Die Restricted ID wird im Interesse des Benutzers (zur Wiedererkennung) und nur durch seine Mitwirkung sowie durch den Einsatz des Berechtigungszertifikats (und dem korrespondierenden privaten Schlüssel) erhoben und ist daher als unkritisch einzustufen. Sie sollte aus Sicherheitsgründen aber nicht für alle Benutzer im LDAP-Verzeichnis lesbar sein (entsprechend der Richtlinie bzgl. transformierter Passwörter).

4.7.3. Datenzugriff

Auf den Nachnamen und Vornamen der Benutzer haben alle anderen Benutzer am Institut Zugriff, da diese für viele Dienste benötigt werden.

Auf die restlichen Daten darf nur die Rechnerbetriebsgruppe Zugriff haben, was durch entsprechende Datenschutzrichtlinien gewährleistet ist. Hier ist es allerdings ausreichend, wenn zwei Personen (hinsichtlich Krankheits- und Urlaubsvertretung) auf die Adressdaten zugreifen können – beispielsweise zur Erfüllung der Auskunftspflicht gegenüber staatlichen Behörden (siehe 4.4.3).

4.8. Sicherheit der Dienstanwendung

Für die Betrachtung der Sicherheitsgefährdungen und den daraus abzuleitenden Sicherheitsmaßnahmen ist es sinnvoll, sich an den Komponenten der Dienstanwendung zu orientieren. Bei der Anbindung an den eID-Service (der Bundesdruckerei) gelten die gleichen Gefährdungen und Maßnahmen, wie sie im Abschnitt 3.3.6 dargelegt wurden.

Beim LDAP-Server sind zwei Aspekte besonders zu berücksichtigen:

1. der Zugriff auf die Objekte des DIT und deren Attribute,
2. die Verfügbarkeit des Verzeichnisdienstes.

Da bereits LDAP-Server am Institut für Informatik betrieben werden und zudem zahlreiche Literatur existiert (z. B. [LU09], Abschnitt 3.7), wie man den Zugriff auf ein DIT regelt, empfiehlt es sich, die bestehenden Regeln zu übernehmen. Dennoch sollten die Betrachtungen zum Datenschutz in Abschnitt 4.7 berücksichtigt werden und in das Regelwerk einfließen.
4.8. Sicherheit der Dienstanwendung


Da durch die Webanwendung direkt Accounts der Organisation modifiziert werden können, ist ein Auditing der im Backend ausgeführten Aktionen (LDAP-Zugriffe, etc.) empfohlen und kann leicht als zusätzliches Backend-Modul oder als Proxy vor den LDAP-Zugriffen implementiert werden.

5. Implementierung

Dieses Kapitel beschreibt Werkzeuge und Gesichtspunkte, die für die Implementierung des im vorangegangenen Kapitel präsentierten Benutzerverwaltungsdienstes relevant waren. Um die Auswahl der Software-Komponenten nachvollziehen zu können, werden zunächst die Entscheidungskriterien erläutert und im Anschluss einzelne Komponenten beschrieben.

5.1. Abhängigkeiten und Auswahlkriterien


Wie bereits angemerkt wurde, soll für die Realisierung des Prototyps der eID-Service der Bundesdruckerei angebunden werden. Die Bundesdruckerei stellt hierzu ein Servlet namens eidconnector zusammen mit ausführlicher Dokumentation [BDr10] zur Verfügung. Ein Servlet ist eine spezielle Java-Klasse, die für die Belange einer Webanwendung passende Schnittstellen implementiert. Dazu gehören beispielsweise Methoden zur Verarbeitung der HTTP-Methoden GET, POST, etc.


Durch diese Abhängigkeiten kann die Implementierung nur in einer Programmiersprache erfolgen, für die es einen Compiler bzw. Interpreter gibt, der Zielcode für die JVM erzeugt. Dazu zählen, neben Java, beispielsweise die Sprachen: Groovy, JRuby, Scala, Jython. Die Implementierung des Prototyps erfolgte in Jython.
5. Implementierung


Ferner soll für die Implementierung des Prototypen (nach Möglichkeit) nur freie Software benutzt werden, um allen Interessierten die Möglichkeit zu geben, einen gleichartigen Dienst auf Basis der vorgestellten Dienstanwendung testen und einsetzen zu können.

5.2. Komponenten des Dienstes und Werkzeuge

Nachfolgend soll das Setup der benutzten Rechner-Instanzen kurz vorgestellt werden, wie es für die Demonstration des Prototyps benutzt wird.

5.2.1. Virtuelle Maschinen zum Aufbau der Demonstrationsumgebung


5.2.2. VM I – Server-Dämonen

Der „Hauptserver“ beheimatet bis auf das Testlogin-SSH-System alle benötigten Dämonen zum Betrieb des Dienstes, d. h., sofern keine Test-Logins demonstriert werden sollen, reicht dieser Server aus. Als Nachweis der Backend-Aktionen könnte auch ein LDAP-Browser zum Betrachten der angelegten und modifizierten LDAP-Objekte dienen. Für die Software-Installation wurden, sofern möglich, Pakete der Linux-Distribution benutzt. Dies hat den Vorteil, dass über ein zentrales Kommando auf jedem Server Software-Updates automatisch und entsprechend den ausgewählten Paketen installiert werden können, ohne dass der Systembetreuer die Software hinsichtlich der Veröffentlichung von Sicherheits-
5.2. Komponenten des Dienstes und Werkzeuge


Anwendungsserver


Webserver

Als Webserver an der Benutzerschnittstelle wird der Apache-Webserver eingesetzt. Die Standardmethode zur Prozesskommunikation mit dem Tomcat-Server ist die Verwendung des Apache-Plugins mod_jk, das wegen einer Besonderheit allerdings hier nicht eingesetzt wird. Abschnitt 5.3.5 am Ende dieses Kapitels geht auf die Besonderheit ein.

LDAP-Server

Um die Anbindung des LDAP-Verzeichnisses nicht mit dem Produktivsystem des Instituts für Informatik testen zu müssen, wird ein (minimaler) OpenLDAP-Server für dem Prototyp betrieben. Dabei werden nur die Standard-Schemata

- core.schema,
- cosine.schema,
- nis.schema,
- inetorgperson.schema,
- samba.schema aus dem Paket samba-doc
Um die grundlegende Verzeichnisstruktur in den (noch leeren) DIT zu laden, wurde die LDIF-Datei B.1 in Anhang B, zusammen mit dem Programm ldapadd benutzt.

**SSH-Server**

Der „Hauptserver“ erhält für die Konfiguration der Dämonen und für das Debugging einen SSH-Zugang, der vom Entwickler bzw. Administrator des Dienstes genutzt wird.

**5.2.3. VM II – Testlogin-System**


**LDAP-Anbindung**


Zur Überprüfung des Benutzerpassworts wird die PAM-Bibliothek\(^2\) benutzt, in der vorab der LDAP-Server als Authentifikationsmechanismus über ein spezielles Backend-Modul konfiguriert wurde. Dieses Modul baut eine Verbindung zum LDAP-Server auf und versucht eine bind-Operation mit den vom Benutzer eingegebenen Credentials. Ist die Operation erfolgreich gewesen, gilt das auch für die Authentifikation des Moduls.\(^3\) Je nach PAM-Konfiguration ist damit die gesamte Authentifikation positiv (sufficient) oder es werden für den Gesamtstatus noch weitere Module abgefragt (required). Im Fehlerfall, wenn das eingegebene Passwort falsch ist, hängt dieser Status ebenfalls von der PAM-Konfiguration.

---


3. Sofern die bind-Option simple konfiguriert wurde (Standardstellung), werden die Passwörter im Klartext übertragen. Daher ist die Verwendung von TLS zur Transportsicherung (auch als LDAPS bezeichnet) oder alternativen bind-Optionen, wie sie die SASL-Bibliothek ermöglicht, dringend empfohlen.
5.2. Komponenten des Dienstes und Werkzeuge

ab. So gibt es PAM-Module, deren Fehlschlag nicht oder nur bedingt relevant den Gesamt-
status sind (optional bzw. sufficient). In [vJ06], S. 9ff., ist das Verfahren ausführlich dar-
gestellt.

Am Ende des fehlerfreien Loginprozesses erhält der Benutzer seine eingestellte Login-
Shell und kann mit den erteilten Rechten auf dem SSH-Server arbeiten.

Dieser Ablauf verdeutlicht auch die Notwendigkeit der beiden Mechanismen nsswitch
und PAM. Während der nsswitch die Accountdaten bereitstellt, führt PAM ausschließlich
die Authentifikation und, davon abhängig, zusätzliche Aktionen durch – beispielsweise
das Erstellen eines Homeverzeichnisses für einen Nutzer, der sich erstmals am System ein-
loggt. Ohne PAM müssten die (transformierten) Passwörter, wie beim (veralteten) Network
Information System (NIS), vom LDAP-Server für die angeschlossenen UNIX-Rechner expor-
tiert werden, die damit eine (locale) Passwortprüfung durchführen würden. Die daraus
resultierenden Probleme wurden in Abschnitt 2.3.1 beschrieben.

5.2.4. Testclient

Der Testclient dient als Demonstrationsplattform und simuliert einen (fiktiven) Benutzer,
der den Dienst zur Verwaltung seines Accounts aufruft. Als notwendige Software sind
daher (in einer virtuellen Maschine mit Windows XP) die AusweisApp, zwei Webbrowser
(Mozilla Firefox und Microsoft Internet Explorer), ein Kartenlesegerät (Omnikey Card-
Man 5321) und ein SSH-Client zur Verbindung mit dem Testlogin-System installiert wor-
den. Darüber hinaus wird mindestens ein Test-Ausweis benötigt – im Rahmen des An-
wendungstests zum neuen Personalausweis standen bis zum 30.10.2010 Ausweise aus der
„Mustermann-Familie“ zur Verfügung.

5.2.5. Entwickler-PC

Der Entwickler-PC benötigt neben dem obligatorischen Editor (oder einer Entwicklungs-
umgebung) ebenfalls einen Webbrowser (hier: Mozilla Firefox) und einen SSH-Client, um
Installations-, Konfigurations- und Debug-Aufgaben auf den Servern durchführen zu kön-
nen. Im Gegensatz zum Testclient ist hier nicht die Installation der AusweisApp und eines
Kartenlesegerät es erforderlich. Eine eID-Sitzung kann funktional auch mit einem Testdienst
des eID-Service durchgespielt werden, der keines Ausweises bedarf und den statischen Da-
tensatz eines fiktiven Ausweises über die eID-Schnittstelle liefert.

Für den Webbrowser Firefox existiert die Erweiterung (Add-on) namens HttpFox, mit
der sich der Verlauf einer Websitzung, also die HTTP-Aufrufe inklusive der benutzten Pa-
rameter und ausgetauschten Daten, nachvollziehen lässt.

Zusätzlich kann für den Zugriff auf das LDAP-Verzeichnis ein LDAP-Browser benutzt
werden, um direkt Manipulationen am DIT vorzunehmen bzw. Backend-Operationen über-
wachen zu können. Sowohl das Add-on HttpFox als auch der Browser LdapAdmin⁴ haben
sich bei der Entwicklung der Webanwendung bewährt.

⁴http://ldapadmin.sourceforge.net/.
5.3. Die Dienstanwendung

Im Mittelpunkt der Betrachtung steht die Dienstanwendung, deren modularer Aufbau die Komplexität beherrschbar macht und Flexibilität bezüglich zusätzlicher Funktionen bringt. Neue Module können später leicht integriert werden.

5.3.1. Verzeichnisstruktur

Für eine bessere Übersicht werden die verwalteten Dateien thematisch in Unterverzeichnissen gruppiert. Die folgende Tabelle gibt einen Überblick:

<table>
<thead>
<tr>
<th>Pfad</th>
<th>Beschreibung</th>
</tr>
</thead>
<tbody>
<tr>
<td>backend</td>
<td>Backend-Module zum Zugriff auf Subsysteme (z. B. LDAP)</td>
</tr>
<tr>
<td>config</td>
<td>Konfiguration des Dienstes</td>
</tr>
<tr>
<td>eid</td>
<td>Modul zur Anbindung des eID-Servers</td>
</tr>
<tr>
<td>keys</td>
<td>Speicherort für Schlüsseldateien</td>
</tr>
<tr>
<td>lib</td>
<td>Gemeinsam genutzte Klassen und Methoden</td>
</tr>
<tr>
<td>modules</td>
<td>Frontend-Module der Dienstes</td>
</tr>
<tr>
<td>static</td>
<td>Statische Inhalte für Frontend-Module (z. B. Bilder, Stylesheets)</td>
</tr>
<tr>
<td>templates</td>
<td>Vorlagen für Frontend-Module (Trennung von Inhalt &amp; Form)</td>
</tr>
<tr>
<td>index.jsp</td>
<td>Einstiegsseite des Servlets (Session-Initiierung)</td>
</tr>
<tr>
<td>index.py</td>
<td>Startseite des Dienstes (Modul-Auswahl)</td>
</tr>
</tbody>
</table>

5.3.2. Module


**Backend-Module** hingegen werden durch die Frontend-Module mittels `import`-Anweisung eingebunden und nicht als Servlet angelegt. Neue Module diesen Typs können durch Kopieren in das Verzeichnis *backend* installiert werden und müssen entsprechend im gewünschten Frontend-Modul eingebunden werden.

Das **eID-Modul** – im Grunde auch ein Backend-Modul – hat eine Sonderrolle in der Dienstanwendung. Durch ein eigenes Unterverzeichnis und wegen des Import-Mechanismus von Python/Jython ist es möglich, die konkrete Implementierung der eID-Schnittstelle austauschen zu können, da zunächst versucht wird, die Datei *__init__.py* im Verzeichnis *eid* zu laden. Diese Datei kann dann auf die gewünschte eID-Implementierung durch einen erneuten Import der Modul-Bezeichner (`from bdr import eid`) verweisen, ohne dass ein

\(^5\)Die Module in diesem Verzeichnis werden an der Dateiendung * .py* identifiziert.
5.3. Die Dienstanwendung

Frontend-Modul angepasst werden muss. Dazu bedarf es aber einer genauen Schnittstel-
len-Definition der Methoden, die vom eID-Modul bereitgestellt werden. Durch den Aufruf
der Methode `sendRequest()` des eID-Moduls wird die eigentliche eID-Sitzung gestartet.
Im Falle des Bundesdruckerei-eID-Services wird also ein SAML-Request an den eID-Server
geschickt usw. Nachdem der Personalausweis gelesen wurde, erfolgt durch den Webbrowser
 des Nutzers der (automatische) Aufruf der Callback-URL (auch `assertionConsumerUrl`).
In dieser Anwendung wird dazu ein Servlet auf Basis von `verarbeiteAusweisdaten.py`
eingerichtet, das zunächst (anhand eines Session-Cookies) prüft, ob die Sitzung der Weban-
wendung gültig ist, und lässt dann bei erfolgreicher Prüfung die SAML-Response, durch
Aufruf der Methode `handleResponse()` des eID-Moduls, auswerten. Abschließend wird
die `callback()`-Methode des ursprünglichen Frontend-Moduls zur Verarbeitung der Aus-
weisdaten gerufen. Welche Methode konkret zu rufen ist, ermittelt das Servlet anhand der
in einer Session-Variablen hinterlegten Referenz.

5.3.3. eID-Anbindung

Eigens um die Implementierung von eID-Anbindungen zu erleichtern und Probleme auf
Clientseite auszuklammern, bietet der eID-Service der Bundesdruckerei zwei Dienstaus-
prägungen an:

1. Der Testservice (Sandbox) liefert einen statischen Datensatz, ohne einen nPA auslesen
zu müssen.

2. Der Liveservice (Referenzsystem) ermöglichte während des Anwendungstests zum
neuen Personalausweis das Auslesen von Testausweisen. Eine ähnliche Funktion bie-
tet die Bundesdruckerei auch nach dem Ablauf des Anwendungstests für spezielle
Ausweismuster („Test Online-Ausweisfunktion“) an. Der operative Betrieb zum Aus-
lesen von „echten“ Personalausweisen (inklusive der nötigen PKI-Strukturen) wird
hingegen durch einen separaten eID-Server (Wirksystem) angeboten.

Mit dem Testservice lassen sich zudem Fehlersituationen simulieren. Nach Anpassung
der Dienst-URL des eID-Servers (`eIdServiceDestination`) liefert der Dienst pro eID-Sitzung
eine andere Fehlermeldung, so dass die eigene Anwendung auf die korrekte Fehlerbehand-
lung hin untersucht werden kann.

5.3.4. Konfiguration

Das Verzeichnis `config` beinhaltet Konfigurationsdateien für die Module der Anwendung.
Mittels der Python-Klasse `ConfigParser` aus der Standardbibliothek ist es leicht möglich,
anwendungsspezifische Parameter zu definieren und in den Modulen zu verwenden. Die
Syntax der Konfigurationsdateien ist an die unter Windows gebräuchlichen INI-Dateien
angelehnt. Auf diese Art werden vom Prototyp vor allem Einstellungen für den Zugriff
auf das LDAP-Verzeichnis (siehe `ldap.cfg` im Anhang B.2) und auf den eID-Server (siehe
testserver.cfg und liveservice.cfg im Anhang B.2) festgelegt.

6 Alternativ kann ein Nutzer auch auf Basis eines SAML-Attributs wiedererkannt werden, sofern der Web-
browser keine Cookies übermittelt.
7 http://docs.python.org/library/configparser.html
5. Implementierung

Typische Anpassungen

Nach der Installation der Dienstanwendung sind typischerweise folgende Anpassungen der Konfiguration vorzunehmen:

- Es ist Schlüsselmaterial für den Zugriff auf den eID-Service gemäß den Angaben des eID-Providers zu erzeugen und im Verzeichnis `keys` abzulegen, vgl. [BDr10], Abschnitt 4.2.


- Bei Nutzung des Bundesdruckerei-eID-Service sind in der Datei `liveservice.cfg` folgende Parameter anzupassen:
  - `serviceProviderIssuerUrl` – die Einstiegsseite (Mandantenkennung) des Dienstes,
  - `assertionConsumerUrl` – die Adresse die nach dem Auslesen des Ausweises vom Client aufgerufen wird (Callback-URL),
  - Angaben zum Schlüsselmaterial (Dateinamen, Zugriffskennwörter auf private Schlüssel).

5.3.5. Besonderheiten bei der Implementierung

Webserver zum Umschreiben der URLs

Während des Anwendungstests bis zur Veröffentlichung von Version 1.5 des eidconnectors war es nicht möglich, die Callback-URL des eigenen Dienstes zur Laufzeit zu konfigurieren und (über den SAML-Request) dem eID-Server mitzuteilen. Beim Liveservice wurde daher die `AssertionConsumerUrl` statisch für den Mandanten in der Datenbank des eID-Servers hinterlegt und während der Erzeugung der SAML-Response integriert. Das Problem an dieser Implementierung stellte sich wie folgt dar: Der Testservice schickte die SAML-Response immer an die vordefinierte URL:

```
https://localhost:1443/Example-ServiceProvider/saml/SAMLAssertionConsumer
```

Dies erschwerte zum einen die Demonstration, weil zunächst ein TCP-Tunnel vom lokalen Rechner (Port 1443) zum Anwendungsserver, beispielsweise per SSH, aufgebaut werden musste. Zum anderen musste aufgrund der in Abschnitt 5.2.2 beschriebenen Erzeugung der URL-Pfade das Callback-Servlet der eigenen Webanwendung in der Testphase stets dem Schema `Example-ServiceProvider/saml/SAMLAssertionConsumer` entsprechen.

Als Lösung dieses Problems wurde deshalb der Apache-Webserver dem Tomcat-Anwendungsserver vorgeschaltet, der per Erweiterungsmodul `mod_proxy` die Callback-URL bei Nutzung des Testservice so umschreibt, dass es für den Tomcat-Server so aussieht, als wurde die selbst definierte URL des Benutzeroberstaltungsdienstes aufgerufen.  

8Nur der eID-Server kann aufgrund des Kommunikationsablaufs bestimmen, an welche URL die SAML-Response durch den Nutzer geschickt wird.

9Da diese Umschreibung jedoch erst beim Dienst stattfand, musste der (Test-)Nutzer dennoch wie beschrieben einen Tunnel zum Anwendungsserver aufbauen. Erst seit Version 1.5 des eidconnectors ist eine „reguläre“ Nutzung des Testservice mit der eigenen Anwendung möglich.
5.3. Die Dienstanwendung

LDAP-Anbindung mit Jython

Im Gegensatz zu anderen Funktionen, für die man sich bei Python-Modulen der Standardbibliothek bedienen kann, ist der Zugriff auf LDAP-Server etwas schwieriger zu realisieren. Für die CPython-Laufzeitumgebung existiert dafür das separate Modul python-ldap\(^{10}\), das aber nicht für Jython zur Verfügung steht. An dieser Stelle wird deshalb das Java Naming and Directory Interface (JNDI) benutzt, welches eine LDAP-Unterstützung ermöglicht und mit der Java-Standardbibliothek ausgeliefert wird.

Das Backend-Modul ldap.py kapselt diese Funktionalität in der Klasse ldapConnection. Die Verbindungsparameter werden dazu aus der Konfigurationsdatei ldap.cfg gelesen.

5.3.6. Quelltexte

Die Quelltextdateien der Dienstanwendung befinden sich auf der CD-ROM, die den gedruckten Exemplaren dieser Arbeit beiliegt.

\(^{10}\)http://www.python-ldap.org/.
6. Fazit

6.1. Zusammenfassung


Abschließend werden Hinweise und Empfehlungen hinsichtlich der Implementierung eines solchen Dienstes am Beispiel des für diese Arbeit entwickelten Prototyps abgeleitet.
6. Fazit

6.2. Ausblick

Die Arbeit hat gezeigt, dass die Hauptaufgabe „Zurücksetzen eines Passworts“ mit Hilfe des neuen Personalausweises effektiv umgesetzt werden kann, sofern die Anbindung an einen eID-Server vollzogen und der Ausweis initial im IT-System der Organisation erfasst und mit dem entsprechenden Benutzerkonto (Useraccount) verknüpft wurde.


1 Eine Online-Einschreibung in Form einer Webanwendung haben offenbar viele deutsche Hochschulen eingeführt, so dass eine Anpassung dieser Anwendung durchaus möglich erscheint.
A. Screenshots der Webanwendung

A.1. Startseite: Modulauswahl

Abbildung A.1.: Modulauswahl.
A. Screenshots der Webanwendung

A.2. Test-Modul

Test-Modul

Diese kleine Test-Anwendung soll das Ändern folgender Attribute aus dem neuen Personalausweis (mPA) ermöglichen:

- Vorname
- Name
- Ausdruck
- Adressenspezifisches Kennzeichen (Restricted ID)

Abbildung A.2.: Daten auslesen starten.

Test-Modul

Diese Daten konnten aus dem Ausweis gelesen werden:

- placeOfResidence
- state: Bayern
- street: Musterpfad 1b
- city: München
- zipcode: 10107
- country: DE
- givenName: Kai
- familyName: Mustermann
- restrictedId X3y3k+hL2VpPt9EX04aA7S7UJ22vFw77X9x0U=

Abbildung A.3.: Anzeige der „Ausweisdaten“. 
A.3. Ein neues Passwort setzen

Neues Passwort

Falls Sie das Passwort Ihres Accounts am Institut für Informatik vergessen haben, wählen Sie ein neues Passwort und tragen es in das folgende Formular jeweils in die entsprechenden Felder ein.

Im Anschluss wird eine Verbindung zu Ihrem Personalausweis aufgebaut und es werden Daten zur Wiedererkennung Ihres Ausweises und zur Erstellung Ihres Accounts ausgelesen. Bitte stellen Sie sicher, dass alles notwendige vorliegt und eingeschlossen ist (Kartenleser, Biogesicht/AusweisApp, neuer Personalausweis).

Passwort:  

Passwort (Wiederholung):  

Starte Authentifikation

Zurück zur Startseite

Abbildung A.4.: Eingabe des neuen Passworts.
A. Screenshots der Webanwendung

Abbildung A.5.: Bestätigung der Datenübermittlung.

Abbildung A.6.: Status der Transaktion und Anzeige des Nutzerkennzeichens.
A.4. Einen neuen Account beantragen.

Abbildung A.7.: Ausfüllen des Formulars.
A. Screenshots der Webanwendung

Abbildung A.8.: Bestätigung der Datenübermittlung.

Abbildung A.9.: Status der Transaktion und Anzeige des Nutzerkennzeichens.
A.5. Fehlermeldung

Abbildung A.10.: Der Ausweis wurde bereits mit einem Benutzerkonto verknüpft.
B. Konfigurationsdateien

B.1. LDAP

Listing B.1: Initiale LDAP-Einträge: informatik.ldif.

```plaintext
1 dn: dc=informatik,dc=hu-berlin,dc=de
2 objectClass: top
3 objectClass: dcObject
4 objectClass: organization
5 dc: informatik
6 o: Institut für Informatik

7 dn: ou=all,dc=informatik,dc=hu-berlin,dc=de
8 objectClass: top
9 objectClass: organizationalUnit
10 ou: all

11 dn: ou=People,ou=all,dc=informatik,dc=hu-berlin,dc=de
12 objectClass: top
13 objectClass: organizationalUnit
14 ou: People
15 description: Benutzer-Objekte

16 dn: uid=student,ou=People,ou=all,dc=informatik,dc=hu-berlin,dc=de
17 objectClass: top
18 objectClass: person
19 objectClass: organizationalPerson
20 objectClass: inetOrgPerson
21 objectClass: posixAccount
22 objectClass: shadowAccount
23 objectClass: sambaSamAccount
24 uid: student
25 shadowLastChange: 1234567890
26 shadowFlag: 0
27 sambaPwdLastSet: 1234567890
28 sambaAcctFlags: [U ]
29 sambaSID: S-1-5-21-1234567890-1234567890-1234567890-12345
30 givenName: Test
31 sn: Student
32 uidNumber: 10001
33 gidNumber: 20001
34 gecos: Test Student, Informatik, CMS-1234
35 homeDirectory: /home/student
36 loginShell: /bin/bash
37 mail: test.student@informatik.hu-berlin.de
38 businessCategory: Informatik, CMS-1234
```

79
B. Konfigurationsdateien

Listing B.2: OpenLDAP-Client-Bibliothek: /etc/ldap/ldap.conf.

```
# LDAP Defaults
#
# See ldap.conf(5) for details
# This file should be world readable but not world writable.

BASE ou=all,dc=informatik,dc=hu-berlin,dc=de
URI ldaps://ldap.example.com/
TLS_CACERT /etc/ssl/certs/ldap.example.com.pem
```


```
# @(#) $Id: ldap.conf,v 2.48 2008/07/03 02:30:29 lukeh Exp $
#
# This is the configuration file for the LDAP nameservice
# switch library and the LDAP PAM module.
#
# PADL Software
# http://www.padr.com
#
# Your LDAP server. Must be resolvable without using LDAP.
# Multiple hosts may be specified, each separated by a
# space. How long nss_ldap takes to failover depends on
# whether your LDAP client library supports configurable
# network or connect timeouts (see bind_timelimit).
# host 127.0.0.1
#
# Another way to specify your LDAP server is to provide an
uri ldaps://ldap.example.com/
```

Listing B.4: PAM-LDAP-Modul: /etc/pam_ldap.conf.

```
uri ldaps://ldap.example.com/
ldap_version 3
pam_password_crypt
```

Listing B.5: PAM-Konfiguration: /etc/pam.d/common-auth.

```
auth sufficient pam_ldap.so debug
auth required pam_unix.so nullok_secure
```
B.2. Benutzerverwaltungsdienst

Listing B.6: Konfiguration config/testservice.cfg für das eID-Modul.

```plaintext
[DEFAULT]
keyLocation = keys

[general]
# Url des eID-Service
eIdServiceDestination = https://test.eid-service.de/epa
serviceProviderIssuerUrl = https://localhost:1443/Example-ServiceProvider
assertionConsumerUrl = https://npa.informatik.hu-berlin.de/account/
                      verarbeiteAusweisdaten.py
keyType = PEM

[PEM-KEYS]
signatureKeyPassword = TopSecretPrivateKeyPassPhrase
# Name der Datei mit dem privaten Signaturschluessel
signatureKeyLocation = %(keyLocation)s/testservice/MySignPrivateKey.pem
# Passwort zur Verwendung des privaten Verschlüsselungsschluessel
encryptionKeyPassword = TopSecretPrivateKeyPassPhrase
# Name der Datei mit dem privaten Verschlüsselungsschluessel
encryptionKeyLocation = %(keyLocation)s/testservice/MyEncrPrivateKey.pem

[eIdServiceCerts]
eIdServiceSigCertificate = %(keyLocation)s/eIdServiceSign.cer
eIdServiceEncCertificate = %(keyLocation)s/eIdServiceEncr.cer
```

Listing B.7: Konfiguration config/liveservice.cfg für das eID-Modul.

```plaintext
[DEFAULT]
keyLocation = keys

[general]
# Url des eID-Service
eIdServiceDestination = https://live.eid-service.de/epa
serviceProviderIssuerUrl = https://npa.informatik.hu-berlin.de/account/
assertionConsumerUrl = https://npa.informatik.hu-berlin.de/account/
                      verarbeiteAusweisdaten.py
keyType = PEM

[PEM-KEYS]
signatureKeyPassword = TopSecretPrivateKeyPassPhrase
# Name der Datei mit dem privaten Signaturschluessel
signatureKeyLocation = %(keyLocation)s/liveservice/MySignPrivateKey.pem
# Passwort zur Verwendung des privaten Verschlüsselungsschluessel
encryptionKeyPassword = %(signatureKeyPassword)s
# Name der Datei mit dem privaten Verschlüsselungsschluessel
encryptionKeyLocation = %(keyLocation)s/liveservice/MyEncrPrivateKey.pem

[eIdServiceCerts]
eIdServiceSigCertificate = %(keyLocation)s/eIdServiceSign.cer
eIdServiceEncCertificate = %(keyLocation)s/eIdServiceEncr.cer
```
B. Konfigurationsdateien

Listing B.8: Aufbau eines Beispiel-Frontend-Moduls modules/example.py.

```python
import ...

# Modul-Beschreibung fuer Einstiegseite der Accountverwaltung
class Meta:
    name = 'FrontendModul'
    title = 'Titel im Browserfenster'
    description = 'Modul-Beschreibung auf der Startseite'
    # Position des Modul-Eintrags auf der Startseite
    ordering = 1

class example(AccountModule):
    def init(self):
        # Hier erfolgt die Initialisierung des Frontendmoduls,
        # u.a. die Einrichtung der eID-Schnittstelle.
        ...

    def doGet(self, response, request):
        # Hier wird der abgebildete Prozess des Moduls gestartet,
        # i.d.R. durch Präsentation eines HTML-Formulars.
        # (Es hat sich bewährt hierfür eine Template-Software zu verwenden.)
        ...

    def doPost(self, response, request):
        # Die Eingabe-Daten des Formulars aus doGet() werden hier verifiziert
        # und die Übergabe der "Ablaufkontrolle" an die eID-Komponente
        # vorbereitet.
        ...

    def callback(self):
        # Diese Methode wird vom "Rücksprung-Programm" aufgerufen und
        # somit eine Weiterverarbeitung des Ausweisdaten durch dieses Modul.
        # Es wird in dieser Arbeit das Rücksprungprogramm "
        # verarbeiteAusweisdaten.py"
        # genutzt, welches die Modul-Auswahl anhand einer Sitzungsvariable
        # trifft.
        ...
```

Listing B.9: Konfiguration config/ldap.cfg für das LDAP-Backend-Modul.

```python
[LDAP]
serverUrl = ldap://localhost/
baseDn = ou=nPAPeople,ou=all,dc=informatik,dc=hu-berlin,dc=de
bindDn = cn=admin,dc=informatik,dc=hu-berlin,dc=de
bindPw = secret
```
Listing B.10: Tomcat-Konfiguration `web.xml` der Webanwendung.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.5"
xmlns="http://java.sun.com/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xm/ns/javaee/web-app_2_5.xsd">
  <display-name>Example-ServiceProvider</display-name>
  <servlet>
    <servlet-name>PyServlet</servlet-name>
    <servlet-class>org.python.util.PyServlet</servlet-class>
    <load-on-startup>1</load-on-startup>
  </servlet>
  <servlet-mapping>
    <servlet-name>PyServlet</servlet-name>
    <url-pattern>*.py</url-pattern>
  </servlet-mapping>
  <welcome-file-list>
    <welcome-file>index.jsp</welcome-file>
  </welcome-file-list>
</web-app>
```
### Abbildungsverzeichnis

<table>
<thead>
<tr>
<th>Kapitel</th>
<th>Seite</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1. Aufbau eines neuen Personalausweises, Quelle: Personalausweisportal</td>
<td>21</td>
</tr>
<tr>
<td>3.2. Kommunikationsbeziehungen bei der eID-Anwendung (vereinfacht, modifiziert aus [TR-03127])</td>
<td>24</td>
</tr>
<tr>
<td>3.3. Zugriffskontrolle und Authentisierung während einer eID-Sitzung, modifiziert aus [BKMN08]</td>
<td>26</td>
</tr>
<tr>
<td>3.4. Public-Key-Infrastrukturen für den neuen Personalausweises, modifiziert aus [BKMN08]</td>
<td>30</td>
</tr>
<tr>
<td>3.5. Komponenten beim Dienstanbieter (Quelle: [TR-03130])</td>
<td>31</td>
</tr>
<tr>
<td>3.6. Schnittstellen eines eID-Servers (modifiziert aus [TR-03130])</td>
<td>32</td>
</tr>
<tr>
<td>3.7. Sequenzdiagramm: Funktionaler Ablauf einer eID-Anfrage (Quelle: [TR-03130])</td>
<td>36</td>
</tr>
<tr>
<td>3.8. Einbindung des eID-Server der Bundesdruckerei beim Anwendungstest</td>
<td>37</td>
</tr>
<tr>
<td>3.9. Szenario mit eCard-API-Proxy</td>
<td>40</td>
</tr>
<tr>
<td>4.1. Ein Ebenen-Modell für das Identitätsmanagement (Quelle: [MA07])</td>
<td>42</td>
</tr>
<tr>
<td>4.2. Benutzer-Objekt, das den Autor im LDAP-Verzeichnis des Instituts repräsentiert</td>
<td>45</td>
</tr>
<tr>
<td>4.3. Komponenten des Benutzerverwaltungsdienstes</td>
<td>51</td>
</tr>
<tr>
<td>A.1. Modulauswahl</td>
<td>71</td>
</tr>
<tr>
<td>A.2. Daten auslesen starten</td>
<td>72</td>
</tr>
<tr>
<td>A.3. Anzeige der „Ausweisdaten“</td>
<td>72</td>
</tr>
<tr>
<td>A.4. Eingabe des neuen Passworts</td>
<td>73</td>
</tr>
<tr>
<td>A.5. Bestätigung der Datenübermittlung</td>
<td>74</td>
</tr>
<tr>
<td>A.6. Status der Transaktion und Anzeige des Nutzerkennzeichens</td>
<td>74</td>
</tr>
<tr>
<td>A.7. Ausfüllen des Formulars</td>
<td>75</td>
</tr>
<tr>
<td>A.8. Bestätigung der Datenübermittlung</td>
<td>76</td>
</tr>
<tr>
<td>A.9. Status der Transaktion und Anzeige des Nutzerkennzeichens</td>
<td>76</td>
</tr>
<tr>
<td>A.10. Der Ausweis wurde bereits mit einem Benutzerkonto verknüpft</td>
<td>77</td>
</tr>
</tbody>
</table>
Abkürzungsverzeichnis

ACL...... Access Control List
AES....... Advanced Encryption Standard
AGS....... Amtlicher Gemeindeschlüssel
APDU...... Application Protocol Data Unit
API........ Application Programming Interface
ASCII..... American Standard Code for Information Interchange
BAC........ Basic Access Control
BSI........ Bundesamt für Sicherheit in der Informationstechnik
CA........ Certificate Authority
CA......... Chip Authentication
CAN........ Card Access Number
CHAT....... CHAT Certificate Holder Authorisation Template
CMS........ Computer- und Medienservice
CRL........ Certificate Revocation List
CSCA....... Country Signing Certificate Authority
CVCA....... Country Validating Certificate Authority
DIT......... Directory Information Tree
DFN......... Deutsches Forschungsnetz
DN......... Distinguished Name
DNS......... Domain Name System
DS......... Document Signer
DV......... Document Verifier
EAC......... Extended Access Control
eID......... Elektronische Identität
ePA......... Elektronischer Personalausweis
HSM......... Hardware-Sicherheitsmodul
HTTP....... Hypertext Transfer Protocol
HTTPS...... Hypertext Transfer Protocol Secure
ICAO....... International Civil Aviation Organization
IdM......... Identitätsmanagement
IP......... Internet Protocol
IT......... Informationstechnik
JVM......... Java Virtual Machine
LDAP....... Lightweight Directory Access Protocol
<table>
<thead>
<tr>
<th>Abkürzung</th>
<th>Bedeutung</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAC</td>
<td>Message Authentication Code</td>
</tr>
<tr>
<td>MITM</td>
<td>Man In The Middle</td>
</tr>
<tr>
<td>MRTD</td>
<td>Machine Readable Travel Document</td>
</tr>
<tr>
<td>MRZ</td>
<td>Machine Readable Zone</td>
</tr>
<tr>
<td>nPA</td>
<td>Neuer Personalausweis</td>
</tr>
<tr>
<td>PACE</td>
<td>Password Authenticated Connection Establishment</td>
</tr>
<tr>
<td>PAM</td>
<td>Pluggable Authentication Modules</td>
</tr>
<tr>
<td>PIN</td>
<td>Persönliche Identifikationsnummer</td>
</tr>
<tr>
<td>PKI</td>
<td>Public Key Infrastructure</td>
</tr>
<tr>
<td>POSIX</td>
<td>Portable Operating System Interface</td>
</tr>
<tr>
<td>PSK</td>
<td>Pre-shared key</td>
</tr>
<tr>
<td>PUK</td>
<td>PIN Unblocking Key</td>
</tr>
<tr>
<td>QES</td>
<td>Qualifizierte Elektronische Signatur</td>
</tr>
<tr>
<td>RAM</td>
<td>Random Access Memory</td>
</tr>
<tr>
<td>RFID</td>
<td>Radio Frequency Identification</td>
</tr>
<tr>
<td>rID</td>
<td>Restricted Identification</td>
</tr>
<tr>
<td>SAML</td>
<td>Security Assertion Markup Language</td>
</tr>
<tr>
<td>SSH</td>
<td>Secure Shell</td>
</tr>
<tr>
<td>SSL</td>
<td>Secure Sockets Layer</td>
</tr>
<tr>
<td>SQL</td>
<td>Structured Query Language</td>
</tr>
<tr>
<td>TA</td>
<td>Terminal Authentication</td>
</tr>
<tr>
<td>TAN</td>
<td>Transaktionsnummer</td>
</tr>
<tr>
<td>TCP</td>
<td>Transmission Control Protocol</td>
</tr>
<tr>
<td>TKG</td>
<td>Telekommunikationsgesetz</td>
</tr>
<tr>
<td>TLS</td>
<td>Transport Layer Security</td>
</tr>
<tr>
<td>UID</td>
<td>Unique Identifier</td>
</tr>
<tr>
<td>URL</td>
<td>Uniform Resource Locator</td>
</tr>
<tr>
<td>VM</td>
<td>Virtuelle Maschine</td>
</tr>
<tr>
<td>VPN</td>
<td>Virtual Private Network</td>
</tr>
<tr>
<td>WLAN</td>
<td>Wireless Local Area Network</td>
</tr>
<tr>
<td>WSDL</td>
<td>Web Services Description Language</td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Markup Language</td>
</tr>
</tbody>
</table>
Literaturverzeichnis


Literaturverzeichnis


Alle URLs wurden zuletzt am 31.05.2011 geprüft.
Selbständigkeitsersklärung
Ich erkläre hiermit, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe.

Berlin, den 1. Juni 2011

Unterschrift

Einverständniserklärung
Ich erkläre hiermit mein Einverständnis, dass die vorliegende Arbeit in der Bibliothek des Instituts für Informatik der Humboldt-Universität zu Berlin ausgestellt werden darf.

Berlin, den 1. Juni 2011

Unterschrift