Management Digitaler Identitäten: Aktueller Status und zukünftige Trends

Christian Tietz, Chris Pelchen, Christoph Meinel, Maxim Schnjakin

Technische Berichte Nr. 114

des Hasso-Plattner-Instituts für Softwaresystemtechnik an der Universität Potsdam

Technische Berichte des Hasso-Plattner-Instituts für
Softwaresystemtechnik an der Universität Potsdam

Technische Berichte des Hasso-Plattner-Instituts für
Softwaresystemtechnik an der Universität Potsdam | 114

Christian Tietz | Chris Pelchen | Christoph Meinel | Maxim Schnjakin

Management Digitaler Identitäten

Aktueller Status und zukünftige Trends

Universitätsverlag Potsdam

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.dnb.de/ abrufbar.

Universitätsverlag Potsdam 2017
http://verlag.ub.uni-potsdam.de/

Am Neuen Palais 10, 14469 Potsdam
Tel.: +49 (0)331 977 2533 / Fax: 2292
E-Mail: verlag@uni-potsdam.de

Die Schriftenreihe **Technische Berichte des Hasso-Plattner-Instituts für Softwaresystemtechnik an der Universität Potsdam** wird herausgegeben von den Professoren des Hasso-Plattner-Instituts für Softwaresystemtechnik an der Universität Potsdam.

ISSN (print) 1613-5652
ISSN (online) 2191-1665

Online veröffentlicht auf dem Publikationsserver der Universität Potsdam
URN urn:nbn:de:kobv:517-opus4-103164
http://nbn-resolving.de/urn:nbn:de:kobv:517-opus4-103164

Zugleich gedruckt erschienen im Universitätsverlag Potsdam:
ISBN 978-3-86956-395-4

Um den zunehmenden Diebstahl digitaler Identitäten zu bekämpfen, gibt es bereits mehr als ein Dutzend Technologien. Sie sind, vor allem bei der Authentifizierung per Passwort, mit spezifischen Nachteilen behaftet, haben andererseits aber auch jeweils besondere Vorteile. Wie solche Kommunikationsstandards und -protokolle wirkungsvoll miteinander kombiniert werden können, um dadurch mehr Sicherheit zu erreichen, haben die Autoren dieser Studie analysiert. Sie sprechen sich für neuartige Identitätsmanagement-Systeme aus, die sich flexibel auf verschiedene Rollen eines einzelnen Nutzers einstellen können und bequemer zu nutzen sind als bisherige Verfahren. Als ersten Schritt auf dem Weg hin zu einer solchen Identitätsmanagement-Plattform beschreiben sie die Möglichkeiten einer Analyse, die sich auf das individuelle Verhalten eines Nutzers oder einer Sache stützt.

Ausgewertet werden dabei Sensordaten mobiler Geräte, welche die Nutzer häufig bei sich tragen und umfassend einsetzen, also z. B. internetfähige Mobiltelefone, Fitness-Tracker und Smart Watches. Die Wissenschaftler beschreiben, wie solche Kleincomputer allein z. B. anhand der Analyse von Bewegungsmustern, Positions- und Netzverbindungsdaten kontinuierlich ein „Vertrauens-Niveau“ errechnen können. Mit diesem ermittelten „Trust Level“ kann jedes Gerät ständig die Wahrscheinlichkeit angeben, mit der sein aktueller Benutzer auch der tatsächliche Besitzer ist, dessen typische Verhaltensmuster es genauestens „kennt“.

Wenn der aktuelle Wert des Vertrauens-Niveaus (nicht aber die biometrischen Einzeldaten) an eine externe Instanz wie einen Identitätsprovider übermittelt wird, kann dieser das Trust Level allen Diensten bereitstellen, welche der Anwender nutzt und darüber informieren will. Jeder Dienst ist in der Lage, selbst festzulegen, von welchem Vertrauens-Niveau an er einen Nutzer als authentifiziert ansieht. Erfährt er von einem unter das Limit gesunkenen Trust Level, kann der Identitätsprovider seine Nutzung und die anderer Services verweigern.

Die besonderen Vorteile dieses Identitätsmanagement-Ansatzes liegen darin, dass er keine spezifische und teure Hardware benötigt, um spezifische Daten auszuwerten, sondern lediglich Smartphones und so genannte Wearables. Selbst Dinge wie Maschinen, die Daten über ihr eigenes Verhalten per Sensor-Chip ins Internet funken, können einbezogen werden. Die Daten werden kontinuierlich im Hintergrund erhoben, ohne dass sich jemand darum kümmern muss. Sie sind nur für die Berechnung eines Wahrscheinlichkeits-Messwerts von Belang und verlassen niemals das Gerät. Meldet sich ein Internetnutzer bei einem Dienst an, muss er sich nicht zunächst an ein vorher festgelegtes Geheimnis – z. B. ein Passwort – erinnern, sondern braucht nur die Weitergabe seines aktuellen Vertrauens-Wertes mit einem „OK“ freizugeben.

Ändert sich das Nutzungsverhalten – etwa durch andere Bewegungen oder andere Orte des Einloggens ins Internet als die üblichen – wird dies schnell erkannt. Unbefugten kann dann sofort der Zugang zum Smartphone oder zu Internetdiensten gesperrt werden. Künftig kann die Auswertung von Verhaltens-Faktoren noch erweitert werden, indem z. B. Routinen an Werktagen, an Wochenenden oder im Urlaub erfasst werden. Der Vergleich mit den live erhobenen Daten zeigt dann

an, ob das Verhalten in das übliche Muster passt, der Benutzer also mit höchster Wahrscheinlichkeit auch der ausgewiesene Besitzer des Geräts ist.

Über die Techniken des Managements digitaler Identitäten und die damit verbundenen Herausforderungen gibt diese Studie einen umfassenden Überblick. Sie beschreibt zunächst, welche Arten von Angriffen es gibt, durch die digitale Identitäten gestohlen werden können. Sodann werden die unterschiedlichen Verfahren von Identitätsnachweisen vorgestellt. Schließlich liefert die Studie noch eine zusammenfassende Übersicht über die 15 wichtigsten Protokolle und technischen Standards für die Kommunikation zwischen den drei beteiligten Akteuren: Service Provider/Dienstanbieter, Identitätsprovider und Nutzer. Abschließend wird aktuelle Forschung des Hasso-Plattner-Instituts zum Identitätsmanagement vorgestellt.

To prevent the increasing number of identity thefts, more than a douzend technologies are already existing. They have, especially then authentication with passwords, specific disadvantages or advantages, respectively. The authors of this survey analyzed how to combine these communication standards and protocols to provide more security. They recommend new kinds of identity management systems that are flexible for different user roles and are more convenient to use as the existing systems. As a first step to build such an identity management platform the authors describe how to analyze and use the individual behavior of users or objects.

As a result sensor data of mobile devices are analyzed. Such devices are internet-ready mobiles, fitness tracker and smart watches. Therefore devices that users often carry with them. The researchers describe how these little computers can continously analyze movement patterns, data of location and connected networks and compute a *trust level* from the data. With this trust level, a device can indicate the probability that the current user is the actual owner, because it *knows* the behavioral patterns of the owner.

If the current trust level value (not single biometric data) is send to an external entity like an identity provider, this provider can provide the trust level to all services used by the user. Each service is able to decide which trust level value is necessary for user authentication. If the trust level drops under a this specific threshold the identity provider can deny the access to itself and all other services.

The particular advantages of this identity management approach is that no special and expensive hardware is needed but instead smartphone and *wearables* to evaluate the specific data. Even objects like machines that send data of their own behavior to the internet can be used. The data is continously collected in the background so users do not need to care about it. The data is only used for computing the trust level and never leaves the device. If a user logs into an internet service he does not need to remember a secret anymore, e. g. a password, instead he just needs to give an *OK* to pass on the trust level.

If the user behavior is changing, for example by different movement patterns or unknown or new locations when trying to log into a web services, it can be immediately detected and the access to the smartphone or internet services an be locked for the unauthorized person. In future the evaluation can be extended for example with detecting routines on working days, on weekends or on vacations. The comparisons of learned routines with live data will show if the behavior fits into the usual patterns.

This survey gives a comprehensive overview of techniques in digital identity management and the related challenges. First, it describes different kinds of attack methods which attacker uses to steal digital identities. Then possible authentication methods are presented. Eventually a summary of the 15 most important protocols and technical standards for communication between the three involved players: service provider, identity provider and user. Finally, it introduces the current research of the Hasso-Plattner Institute in the area of identity management.

Inhaltsverzeichnis

Abbildungsverzeichnis

1 Einleitung

Im Internet wächst die Zahl der verfügbaren Dienste rasant. Davon verlangt ein Großteil vom Nutzer eine Registrierung. Diese sind zum Beispiel wichtig für Vertragsabschlüsse – etwa beim Online Einkauf oder bei Einrichtung eines Kontos bei einer Internet-Bank. Dafür werden Daten eingegeben wie E-Mail-Adresse, Wohnort, Anschrift, Geschlecht, Alter usw. Durch Registrierungen können Dienste ihr Angebot auf den Nutzer anpassen. Dieser kann besondere Merkmale auswählen und bekommt passende Angebote im Online-Shop oder durch Werbung. Bei jeder neuen Registrierung auf einer Webseite gibt der Nutzer immer dieselben Daten an, etwa Name oder E-Mail-Adresse. Das ist umständlich und lästig. Bei einer Änderung, wie zum Beispiel der Anschrift nach einem Umzug, müssen die Daten dann bei allen Diensten geändert werden und nicht bloß bei einem, was bequemer wäre.

Die Konsequenz solcher vielen Registrierungen ist, dass ein durchschnittlicher Internetnutzer mehr als 25 Internetkonten hat [12]. Diese Konten werden für diverse Zwecke benötigt wie E-Mailing, Online-Shopping, Online-Banking, Nutzung sozialer Netzwerke, Foren, Spiele, Lernplattformen usw.

Zum Schutz der Konten im Internet werden häufig Benutzername und Passwort verwendet, wobei die Nutzer ihr Passwort selbst bestimmen können. Das bringt einige Sicherheitsprobleme mit sich, mit negativen Folgen für den Schutz der Internetkonten. Denn kurze und einfach merkbare Wörter sind leicht von anderen zu erraten oder können mittels Programmen in allen möglichen Varianten durchprobiert und blitzschnell geknackt werden. Längere und komplexere Passwörter (mit Groß- und Kleinbuchstaben, Ziffern und Sonderzeichen) bieten eine höhere Sicherheit, aber sind kaum im Gedächtnis zu behalten. Das gilt vor allem dann, wenn für jedes Online-Konto ein neues Passwort verwendet werden soll, wie es das BSI rät.[1]

In der Konsequenz verleitet das Nutzer dazu, Passwörter mehrfach zu verwenden. Eine Studie von Florencio and Herley zeigt, dass für 25 Internetkonten 6,5 verschiedene Passwörter verwendet werden [12]. Das heißt: Für je vier Webseiten wird dasselbe Passwort gewählt. Wenn ein Angreifer ein solches Passwort herausbekommen hat, sei es durch Raten, Hacken oder andere Methoden, kann er nicht nur auf ein Konto sondern gleich auch auf mehrere zugreifen und diese für seine Zwecke missbrauchen. Das wird als Identitätsdiebstahl bezeichnet.

[1] `https://www.bsi-fuer-buerger.de/BSIFB/DE/Empfehlungen/Passwoerter/Umgang/umgang.html`, besucht am 16.02.2017.

Laut Statistiken von Risk Based Security, Inc.[2,3], ist es so, dass in den Jahren 2014 und 2015 rund 2 Mrd. Datensätze bei ca. 7.000 Dateneinbrüchen gestohlen wurden. Ungefähr die Hälfte der Datensätze enthalten Passwörter und E-Mail-Adresse (Vgl. Abbildung 1.1 für 2015 aus den Statistiken). Im Internet reichen diese Daten schon aus, um eine Identität stehlen zu können. Auch im ersten Halbjahr von 2016 wurden bereits über 554 Millionen Datensätze gestohlen.[4]

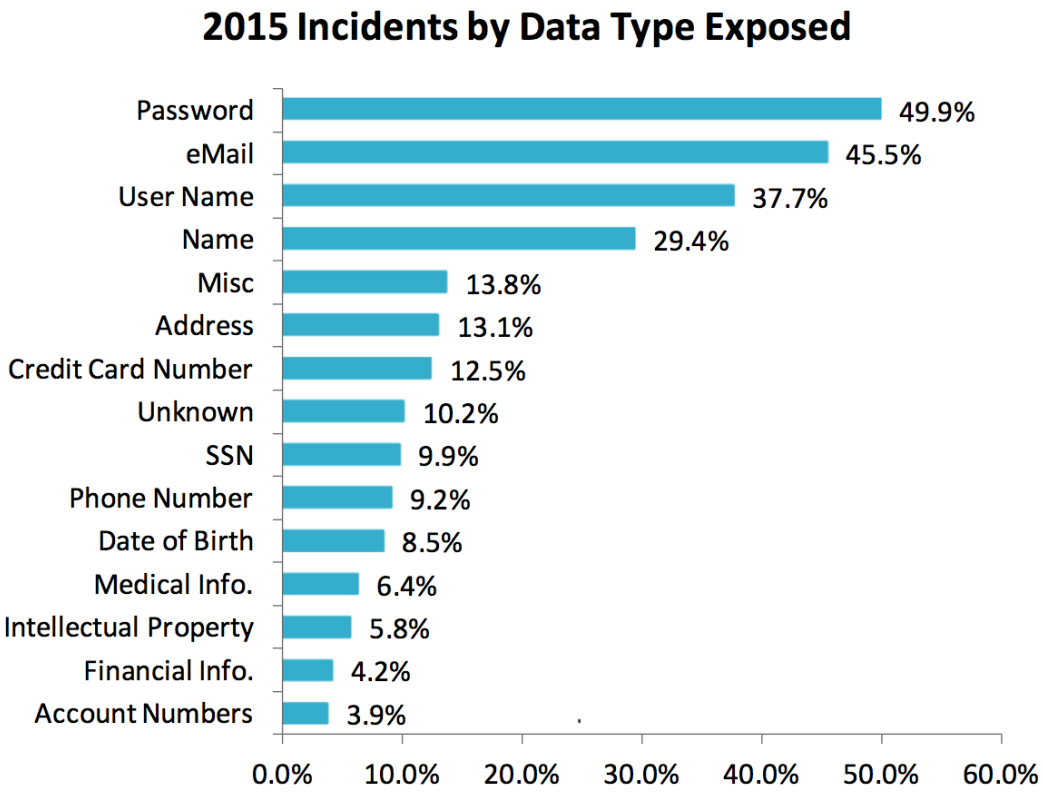

Abbildung 1.1: Arten der gestohlenden Daten aus dem Jahr 2015

Entnommen aus dem Risk Based Security Bericht von 2015

Ein mögliches Szenario für einen Identitätsdiebstahl ist also, dass ein Dieb sich zum Beispiel in den Besitz von Adresse und Kontonummer gebracht hat. Mit diesen Informationen kann er dann bei der betreffenden Bank die Adresse ändern,

[2] `https://www.riskbasedsecurity.com/reports/2014-YEDataBreachQuickView.pdf`, besucht am 16.02.2017.
[3] `https://www.riskbasedsecurity.com/reports/2015-YEDataBreachQuickView.pdf`, besucht am 16.02.2017.
[4] `http://breachlevelindex.com/assets/Breach-Level-Index-Report-H12016.pdf`, besucht am 16.02.2017.

zum Beispiel auf eine Scheinadresse hin. Nach einiger Zeit kann sich der Dieb eine neue Kreditkarte ausstellen und zusenden lassen, mit der das Konto des Nutzers belastet wird.

In Kapitel 2 werden solche allgemeinen Angriffe vorgestellt, mit denen Diebe an Daten der Nutzer kommen können.

Um Diebstähle digitaler Identitäten zu bekämpfen, müssen neue Wege gefunden werden, wie Identitäten vor Angriffen und unbefugten Zugriffen geschützt werden können. Und nicht nur der Zugriff muss geschützt werden, sondern auch das Ändern der Daten ohne erneute Überprüfung. Die Überprüfung selbst sollte dem Nutzer dabei aber keine großen Schwierigkeiten bereiten.

Das sind die hauptsächlichen Probleme, mit dem sich das Identitätsmanagement (IdM) beschäftigt. Allgemein lässt sich sagen: Identitätsmanagement befasst sich damit, Identitäten zu verwalten, sie zu authentifizieren und zu autorisieren – auch über Unternehmensgrenzen hinaus – mit dem Ziel, die Sicherheit und Produktivität zu erhöhen sowie die Kosten und sich wiederholende Aufgaben zu verringern.

1.1 Aufbau der Studie

In den restlichen Abschnitten dieses Kapitels werden Begriffe zur Identität, Ansätze und Anwendungszwecke im Identitätsmanagement erläutert. Das folgende Kapitel 2 behandelt die Möglichkeiten, wie Identiäten gestohlen werden können. Es werden konkrete Angriffsbeispiele vorgestellt. Kapitel 3 stellt Authentifizierungsmethoden dar. Hier wird beschrieben, wie der Nutzer seine Identität nachweisen kann. In Kapitel 4 werden wichtige Protokolle und Technologien vorgestellt, die derzeit existieren. Das 5. Kapitel stellt abgeschlossene und laufende Forschungsprojekte des Hasso-Plattner-Instituts vor. Den Abschluss bildet eine Zusammenfassung mit Ausblick auf die Zukunft des Identitätsmanagements.

1.2 Digitale Identität

Damit ein Anwender einen Internetdienst nutzen kann, muss er sich vorher anmelden bzw. registrieren. Das ist notwendig, damit der Dienst den Nutzer wiedererkennt. Ein Dienst für Online-Shopping muss zum Beispiel wissen: Wer kauft ein, wohin sind die Einkäufe zu liefern, von welchem Konto muss abgebucht werden usw. Der Dienst legt für jeden neuen Nutzer technisch eine eigene *digitale Identität* an.

Eine digitale Identität ist eine Sammlung elektronischer Daten zur Charakterisierung eines Internetnutzers mit einer physischen Identität. Daten, die zu einer digitalen Identität gehören, sind z. B. Nutzername, E-Mail-Adresse, Wohnanschrift, Kontonummer, Passwort usw. und werden als *Attribute* bezeichnet. Ein physischer Nutzer kann sich im Internet mit vielen verschiedenen digitalen Identitäten bewegen (anderer Nutzername, andere E-Mail usw.). Es gibt viele Facetten oder Rollen

der digitalen Identität, die unterschiedliche Attribute benötigen. Die wichtigsten sind „privat“, „staatlich“und „geschäftlich“.

Um den Missbrauch der digitalen Identität zu vermeiden, muss eine Bindung an den jeweiligen physischen Nutzer sichergestellt werden. Die Überprüfung findet in Form eines Anmeldeprozesses bzw. Logins statt. Der Nutzer muss beweisen, dass er Besitzer einer digitalen Identität ist. Das heißt, er muss sich *authentifizieren*. Es gibt viele Möglichkeiten der Authentifikation. Beispiele hierfür sind Passwörter, PIN, Fingerabdruck, Gesichtserkennung, Chipkarten usw. Diese Technologien lassen sich grundsätzlich drei Kategorien zuordnen: Wissen, Besitz und Biometrie. Mehr zu den Authentifizierungsmethoden wird in Kaptiel 3 ausgeführt.

1.3 Ansätze des Identitätsmanagements

Beim Identitätsmanagement kann zwischen vier unterschiedlichen Ansätzen unterschieden werden. Diese lassen sich wiederum zwei verschiedenen Kategorien zuordnen [28].

Der erste Ansatz ist das *isolierte Identitätsmanagement*. Hier verwaltet jeder Dienst seine Nutzer selbst. Das ist der klassische Ansatz, der am weitesten im Internet verbreitet ist. Als Beispiel sind etwa Ebay und Amazon zu nennen.

Zentralisiertes Identitätsmanagement bildet den zweiten Ansatz. Es gibt hier eine zentrale Einheit, die Nutzer verwaltet. Damit können dann mehrere Dienste genutzt werden. Dieser Ansatz lässt sich in Unternehmen finden, in denen es ein zentrales LDAP-Verzeichnis gibt. LDAP steht für Lightweight Directory Access Protocol (dt. leichtgewichtiges Verzeichniszugriffsprotokoll) und bezeichnet einen zentralen Computer, auf dem Dokumente oder auch Anmeldeprofile gespeichert sind und die von überall aus dem Netzwerk abgefragt werden können. Nutzer können sich an einem beliebigen Computer im Netzwerk anmelden und es werden ihre Einstellungen und Daten geladen. Das entsprechende Profil erhält der Computer vom LDAP-Server. Ein solcher Dienst, der Identitätsdaten speichert (hier: Anmeldeprofil) und sie anderen Diensten (hier: Computer im Netzwerk) zur Verfügung stellt, wird auch als Identitätsprovider (IdP) bezeichnet. Das wurde auch auf das Internet übertragen. Beispiele finden sich bereits auf vielen Internetseiten, die etwa einen Button „Login mit Facebook“enthalten. Hier ist dann der Server des sozialen Netzwerks die zentrale Einheit, über die sich alle Nutzer authentifizieren und dann Dienste auf anderen Webseiten, z. B. von Audio- oder Video-Streaming-Anbietern, nutzen können. Nutzer und Dienste müssen hier auf den zentralisierten Identitätsprovider vertrauen.

Der dritte Ansatz nennt sich *dezentralisiertes Identitätsmanagement*. Hier gibt es mehrere Identitätsprovider, die benutzt werden können, um sich bei einem Dienst anzumelden. Ein Beispiel hierfür ist die Anmeldung mit OpenID.[5] Wenn eine Seite OpenID als Anmeldung unterstützt, gibt der Nutzer seine OpenID-URL (Uniform

[5] `http://openid.net`, besucht am 16.02.2017.

Resource Locator, Darstellung einer Webadresse) ein und wird zu seinem Identitätsprovider zwecks Authentifikation weitergeleitet. In Abhängigkeit der URL könnte jeder Nutzer zu einem anderen oder zum selben Identitätsprovider weitergeleitet werden. Dadurch müssen Identitätsprovider dem Dienst im Vorfeld nicht bekannt sein. Am Hasso-Plattner-Institut (HPI) gibt es zum Beispiel einen solchen OpenID-Identitätsprovider.[6] Damit können sich Studenten und Mitarbeiter des HPI auf anderen Webseiten anmelden, indem sie die HPI-OpenID eingeben. Auch hier muss der Dienst den HPI-Identitätsprovider nicht im Voraus kennen, jedoch auf seine Aussage vertrauen.

Der vierte Ansatz besteht im *föderierten (federated) Identitätsmanagement*. Es gibt hier eine Föderation bzw. einen Vertrauenskreis (Circle of Trust) von Diensten und Identitätsprovidern, die gegenseitig ihren Identitätsinformationen vertrauen. Initiativen, die Spezifikationen anbieten, sind WS-Federation[7] und die Kantara Initiative[8] (früher Libertry Alliance). Nutzer des einen Anbieters können dann auch Dienste eines anderen Anbieters innerhalb der Föderation nutzen, ohne sich dort neu zu registrieren oder anzumelden.

Die letztgenannten beiden Ansätze sind sich sehr ähnlich. Der Hauptunterschied besteht darin, wie zwischen zwei Partnern die Vertrauensbeziehung aufgebaut wird. Beim einen geschieht dies spontan bei der Anmeldung, beim anderen entscheiden die Anbieter im Vorfeld darüber.

Die vier Ansätze lassen sich zwei Kategorien zuordnen: dem *domänenbasierten Modell* und dem *offenen Identitätsmodell* (Open Identity Model). Der isolierte und zentralisierte Ansatz gehören zur ersten Kategorie, dem domänenbasierten Modell. Hier dreht sich alles um den einen Account in einer bestimmten Domaine. Im Gegensatz dazu bilden der dezentralisierte und föderierte Ansatz das offene Identitätsmodell. Den dezentralisierten Ansatz findet man häufiger im B2C-Bereich (Business to Consumer), das föderierte Identitätsmanagement ist mehr im B2B-Bereich (Business to Business) anzutreffen [28]. Eine Veranschaulichung der einzelnen Ansätze leistet Abbildung 1.2.

Um digitale Identitäten über Domänengrenzen hinaus zu akzeptieren, bedarf es einer entsprecheden Darstellung der Identität, da der Account in der einen Domäne in der anderen nicht existiert. Zusätzlich soll die Darstellung flexibel sein, da unterschiedliche Dienste unterschiedliche Attribute eines Nutzers benötigen. So reicht es für ein Diskussionsforum aus, nur ein Pseudonym zu haben, während ein Online-Shop zwingend Anschrift und Kontoinformationen benötigt. Aus der Motivation heraus, Identitäten mit anderen zu teilen und zu akzeptieren, entstand das Konzept der so genannten Claims (Behauptungen) und der *Claim-basierten Identität* [5].

[6] `https://openid.hpi.uni-potsdam.de`, besucht am 16.02.2017.
[7] `https://msdn.microsoft.com/en-us/library/bb498017.aspx`, besucht am 16.02.2017.
[8] `https://kantarainitiative.org`, besucht am 16.02.2017.

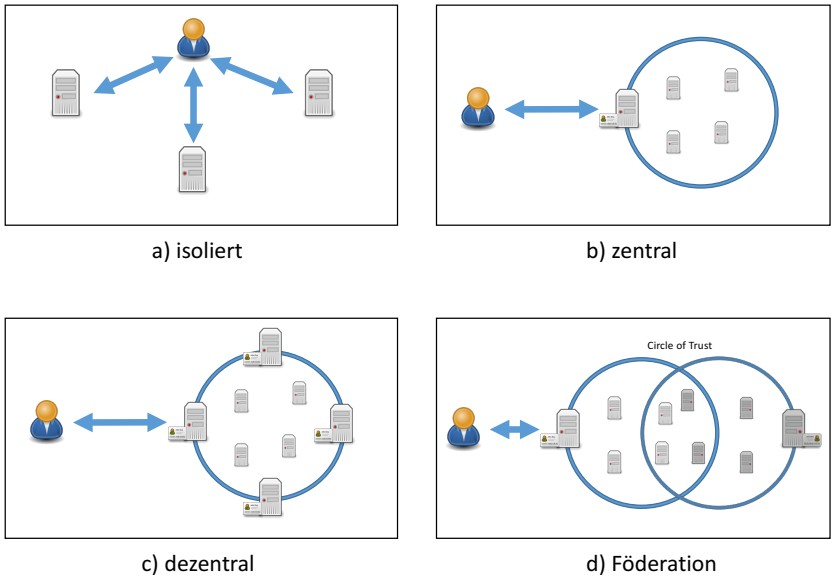

Abbildung 1.2: Die unterschiedlichen Ansätze grafisch dargestellt

1.4 Identitätsmanagement für mehrere Anwendungszwecke

Claim-basierte Identität ist die Art von Darstellung, die für die Kategorie der offenen Identitätsmodelle nötig ist bzw. sie ermöglicht. Behauptungen sind Attribute und machen eine Aussage über eine Entität, zum Beispiel über eine Person. Der Unterschied zwischen Attributen und Claims ist die Art der Nutzung.[9] Der Dienst hat Attribute gespeichert und kann sie bei Bedarf nachschlagen. Behauptungen bzw. Claims werden im Bedarfsfall vom Nutzer oder anderen Diensten an den Ziel-Dienst übermittelt. Der Behauptung wird vertraut, wenn dem Nutzer bzw. dem Absender vertraut wird. Einer Behauptung des Unternehmens eines Nutzers wird mehr Vertrauen geschenkt, als wenn sie vom Nutzer selbst kommt. Claim-basierte Identitäten ermöglichen ein Identitätsmanagement, das mehrere Anwendungszwecke und Implementierungen erlaubt.

Ein Beispiel soll als Veranschaulichung dienen. Setzen wir einmal voraus, dass ein Universitätsinstitut eine Kooperation mit einem Unternehmen eingeht. Mitarbeiter des Instituts dürfen dann Berichte des Unternehmens lesen, aber gleichzeitig soll das Unternehmen nicht wissen, welcher Instituts-Mitarbeiter was gelesen hat.

[9] `https://msdn.microsoft.com/en-us/library/ee517291.aspx`, besucht am 16.02.2017.

Für das Unternehmen ist nur wichtig, dass es ein Mitarbeiter des Instituts ist und nicht jemand anderes. Das Institut bescheinigt die Beschäftigung seines Mitarbeiters. Mit dieser digitalen Bescheinigung (Token) bekommt der Mitarbeiter Zugriff zu den Berichten des Unternehmens. Einer Bescheinigung des Instituts wird mehr Glauben geschenkt als einer Bescheinigung, die sich der Instituts-Mitarbeiter selbst ausstellt.

2 Angriffe auf Identitätsdaten

In diesem Kapitel sollen einige Methoden vorgestellt werden, mit denen Angreifer, Diebe und Betrüger versuchen, an online gespeicherten Daten von Nutzern zu gelangen.

Wie erwähnt, gab es in den Jahren 2014 und 2015 mehr als 2 Mrd. Datendiebstähle. Und fast täglich berichten Medien über weitere gehackte Unternehmensdatenbanken und gestohlene Identitätsdaten. Ein Großteil der entwendeten Daten wird im *dark net*, in den dunklen Ecken des Internets, veröffentlicht oder gewinnbringend veräußert bzw. für weitere illegale Handlungen verwendet oder freigegeben. Diese Datenpreisgaben werden als *Leaks* bezeichnet. Andere Anreize für Cyberkriminelle sind die eigene Profilierung und das Erlangen von Ansehen bei Gleichgesinnten. Die Opfer der Diebstähle, also die Nutzer oder Kunden der geknackten Datenbanken, wissen meistens gar nicht, dass sie von einem Leak betroffen sind. Im ungünstigsten Fall bemerken sie es erst dann, wenn ihre Identitätsdaten bereits missbraucht worden sind.

Ein Fall eines solchen Datenbankangriffs betrifft den Internetkonzern Yahoo. Im September 2016 bestätigte das Unternehmen, dass im Jahr 2014 Daten von mindestens 500 Millionen angemeldeten Nutzern entwendet wurden.[1] Die gestohlenen Daten umfassen neben der E-Mail-Adresse und dem verschlüsselten Passwort auch die Namen, Telefonnummern und Geburtsdaten der User.

Die Angreifer verwenden verschiedene Techniken, um Zugriff auf die Informationen in den Datenbanken zu erhalten. Es wird zwischen zwei grundsätzlichen Arten von Angriffen unterschieden: jene, die sich gegen Datenbanken richten und solche, die sich gegen Nutzer richten, die entweder Zugriff auf die Datenbanken haben – wie beispielsweise Administratoren – oder deren Informationen Teil der Datenbanken sind.

Die am häufigsten verwendete Attacke, die sich direkt gegen Datenbanken richtet, ist die sogenannte *SQL-Injection* (SQL-Einschleusung). Ziel dieses Angriffs ist das Ausnutzen von Sicherheitslücken durch gezielte Manipulation automatisierter Datenbankabfragen. Schnittstelle zwischen Angreifer und Datenbank sind meist die vorhandenen Möglichkeiten für Benutzereingaben. Ein möglicher Angriffspunkt ist beispielsweise das Anmeldefenster für einen Webservice, in das die Nutzer ihre E-Mail-Adresse und das dazugehörige Passwort eingeben. Beim Starten des Anmeldevorgangs wird zunächst überprüft, ob die angegebene E-Mail-Adresse in der Nutzerdatenbank hinterlegt ist, und im Anschluss, ob das

[1] `http://money.cnn.com/2016/09/22/technology/yahoo-data-breach`, besucht am 16.02.2017.

dazugehörige Passwort korrekt ist. Hierfür wird die eingetippte E-Mail-Adresse in eine vorgefertigte Datenbankabfrage eingesetzt. Wenn jedoch vor diesem Schritt nicht überprüft wird, ob es sich um eine E-Mail-Adresse in einem gültigen Format handelt, können die eingebenden Werte die automatisierte Datenbankabfrage so verändern, dass eine völlig andere Art der Abfrage durchgeführt wird. Im schlimmsten Fall gelingt es dem Angreifer, unbeschränkten Administratoren-Zugriff auf das gesamte System und alle gespeicherten Informationen zu erhalten.

Angriffe, die sich gegen die Nutzer von Webservices richten, zielen meist darauf ab, Informationen aus der Kommunikation zwischen dem User und dem Service auszuspähen. So wird bei der Angriffsform *Password Sniffing* der Datenverkehr, also die übermittelten Datenpakete innerhalb eines Netzwerks, überwacht. Hierbei möchte der Angreifer genau die Datenpakete abgreifen, die Login-Daten enthalten. Anhand von Stichworten – wie beispielsweise *login* oder *password* – werden die Datenpakete analysiert und entsprechende Informationen unbemerkt herausgefiltert.

Eine weitere Angriffsform ist das *Password Phishing*. Bei dieser Art von Attacken wird der Nutzer durch gefälschte E-Mails, Webseiten, Kurznachrichten oder Telefonanrufe gezielt getäuscht, damit er von sich aus seine Anmeldedaten oder persönlichen Informationen preisgeben soll. Eine besondere Form des Phishings ist das sogenannte *Social Engineering*. Bei dieser Attacke nimmt der Angreifer direkten persönlichen Kontakt mit der Zielperson auf. So ruft er beispielsweise bei einem Unternehmen an und gibt sich als vermeintlicher Techniker aus. Unter dem möglichen Vorwand, die Mitarbeiter-Accounts überprüfen zu müssen, bittet er die Angestellten um Auskunft über deren Anmeldedaten und kann sich ggf. so Zugang zum System verschaffen.

Eine andere Angriffsform ist das *Password Monitoring*. Hierbei wird mithilfe von kleinen Programmen, so genannte *Keylogger*, jeder Tastaturanschlag unbemerkt im Hintergrund aufgezeichnet. Somit können auch die Anmeldedaten eines Nutzers heimlich registriert werden. Die erfassten Daten werden im Anschluss entweder automatisch an eine E-Mail-Adresse des Angreifers gesendet oder er exportiert die Daten manuell vom lokalen System.

Auch das Raten von Passwörtern kann in schlecht gesicherten Systemen zum Erfolg führen. Für den Administratoren-Zugriff gibt es meist standardisierte Benutzerkonten wie beispielsweise *admin* oder *root*. In diesen Fällen muss der Angreifer lediglich das dazugehörige Passwort herausfinden, da der Benutzername bereits bekannt ist. Üblicherweise wird solchen Accounts bei der Installation eines Systems ein Standard-Passwort zugewiesen, beispielsweise *admin* für den Administratoren-Account. Im Internet lassen sich mit wenigen Klicks lange Listen derartiger Standard-Passwörter aufspüren, die ein Angreifer schnell durchprobieren kann. Sollte das Passwort nach dem Abschluss der Systemerstellung nicht geändert worden sein, hat es ein Angreifer leicht, Zugriff zu erhalten.

Diese verschiedenen Arten von Angriffen verdeutlichen, dass eine sichere Infrastruktur und ein hohes Sicherheitsbewusstsein Grundvoraussetzungen für die Datenspeicherung von sensiblen Informationen sind. Neben den Möglichkeiten, eine Datenbank effektiv vor äußeren Angriffen zu schützen, sind auch alternative

Authentifizierungsmethoden im Auge zu behalten. In Kapitel 3 wird auf solche Verfahren zur Authentifikation eingegangen.

3 Methoden der Authentifikation

Als Authentifikation wird generell der von einer Entität zu erbringende Beweis verstanden, dass ihr eine bestimmte Identität gehört. Im Englischen wird von *authentication* gesprochen. Im Deutschen jedoch werden zwei Begriffe unterschieden: *Authentisierung* und *Authentifizierung*. Die Authentisierung beschreibt den Nachweis einer Person. Der Betreffende muss sich ausweisen und etwas vorlegen, dass seine behauptete Identität bestätigt. Im Gegensatz dazu bezeichnet Authentifizierung die Prüfung der behaupteten Authentisierung. Sie folgt nach der Authentisierung. Ein Nutzer authentisiert sich bei einem Dienst und wird dann von diesem authentifiziert.

Beide Begriffe sollen mit dem Beispiel Online-Shoppingportal illustriert werden. Bei einem solchen Shoppingportal sind Daten von Nutzern gespeichert. Das sind digitale Identitäten. Wenn ein Nutzer diese Webseite benutzen will, muss er angeben, welche gespeicherte Identität zu ihm gehört. Das geschieht durch Eingabe eines Nutzernamens. Damit nicht jedermann sich unter diesem Namen anmelden und den Dienst benutzen kann, muss der Anwender beweisen, dass er wirklich dieser Identität entspricht. Der Beweis wird häufig durch die Eingabe eines Passwortes erbracht. Er authentisiert sich mit seinem Passwort. Danach muss das Portal das Passwort überprüfen. Dadurch authentifziert der Dienst den Nutzer. Es folgt also erst die Authentisierung und dann die Authentifizierung.

Neben der Authentisierung und Authentifizierung gibt es noch die *Autorisierung*. Die Autorisierung beschäftigt sich mit der Überprüfung und Gewährung bestimmter Rechte. Es wird festgelegt, welche Rechte ein Nutzer hat bzw. was er in dem System machen darf (zum Beispiel differenziert danach, ob es sich beim Nutzer um einen Administrator, Moderator oder einen Standard User handelt).

Es gibt grundsätzlich drei Arten, wie eine digitale Identität nachgewiesen werden kann: durch Wissen, Besitz und Biometrie. Im Folgenden werden diese drei Möglichkeiten, auch Faktoren genannt, genauer erläutert und ihre Vor- und Nachteile beschrieben. Danach werden Probleme der einzelnen Varianten genannt und es wird gezeigt, wie sich Kombinationen der Methoden auswirken können.

3.1 Authentifizieren durch Wissen

Wissen-basierte Authentifikation bedeutet, dass ein Nutzer authentifiziert wird, wenn er zeigt, dass er ein bestimmtes Geheimnis kennt, welches nur der rechtmäßige Nutzer wissen kann. Die bekannteste Methode ist ein Passwort oder eine PIN (Persönliche Identifikation-Nummer). Passwörter sind die im Internet am meisten

verbreitete Methode bei der Authentifizierung. Abbildung 3.1[1] zeigt ein typisches Eingabefeld zur Anmeldung mit Passwörtern.

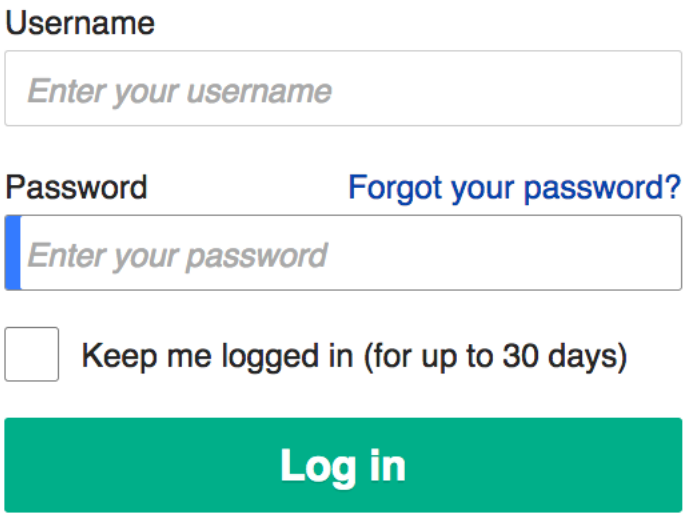

Abbildung 3.1: Typisches Internet-Authentifizierungsformular mit Username und Passwort

Passwörter beschränken sich nicht nur auf Text, sondern können auch grafisch dargestellt sein. Grafische Passwörter [2] werden oftmals in zwei Klassen unterteilt: in Erinnerung und Erkennung [26]. Erinnerungs-basierte (recall-based) grafische Passwörter müssen bei der Eingabe neu produziert werden. Ein Beispiel ist eine (handschriftliche) digitale Unterschrift. Hier muss sich der Nutzer erinnern, wie seine Unterschrift aussieht und diese dann neu schreiben. Ein anderes Beispiel ist die Gesten-Erkennung durch Smartphones. Zur Authentifikation muss der Nutzer die betreffende Geste kennen und sie zeichnen. Auch Text-basierte Passwörter sind Erinnerungs-basiert. Erkennungs-basierte (recognition-based) grafische Passwörter zeigen dem Nutzer zur Authentifikation ein Bild oder eine Reihe von Bildern. Der Nutzer muss dann (wieder-)erkennen, welche Stellen in dem einen Bild das Geheimnis bilden bzw. welche Bilder in der Reihe von Bildern als Passwort festgelegt wurden. Im Bereich der sozialen Netzwerke gibt es zudem die Möglichkeit, Fotos zu verwenden. Zum Beispiel muss der Nutzer alle Bilder anklicken, die einen bestimmten Freund abbilden, oder den Freund benennen, der auf allen Bildern zu sehen ist. Ein solches Foto-basiertes System gibt es beispielsweise bei Facebook zum Verifizieren des Kontos.

[1] `https://en.wikipedia.org/wiki/Password`, besucht am 16.02.2017.

Die Vorteile von Wissen-basierten Authentifizierungen sind, dass sie bereits sehr lange verwendet werden und einfach zu implementieren sind. Viele Nutzer sind mit dieser Methode, vor allem wenn Passwörter verwendet werden, bereits vertraut. Wissen kann von überall aus in der Welt genutzt werden, um die eigene Identität nachzuweisen und Dienste, bei denen man registriert ist, zu benutzen. Wenn Wissen einmal kompromittiert worden sein sollte, kann das Geheimnis schnell und einfach geändert werden.

Die Nachteile von Wissen-basierten Methoden liegen darin, dass Geheimnisse vergessen werden können. Einfache Geheimnisse können von anderen erraten werden. Schriftlich aufgezeichnete Geheimnisse können von anderen gefunden werden. Vor allem Text-basierte Passwörter sind schwer zu merken, wenn sie aus einer gemischten Abfolge von Zeichen, Sonderzeichen und Ziffern bestehen. Grafische Passwörter sind da leichter zu verwenden, weil Menschen sich Bilder viel besser merken können [9]. Allerdings haben auch grafische Passwörter ihre Schwächen. Zum Beispiel können Unterschriften nachgemacht werden. Wenn Fotos verwendet werden, können Freunde ermittelt werden – zum Beispiel in sozialen Netzwerken.

3.2 Authentifizieren durch Besitz

Bei der Besitz-basierten Authentifizierung wird ein Gegenstand geprüft, der einem einzigen registrierten Nutzer gehört. Diese Gegenstände sind Ausweise des Nutzers, müssen also eindeutig sein. Chipkarten sind wohl die bekannteste Möglichkeit der Besitz-basierten Authentifizierung: Nutzer brauchen nur die Karte an ein Lesegerät zu halten und sind dann authentifiziert. Neben Chipkarten können zum Beispiel auch USB-Tokens und Smartphones eingesetzt werden. Es muss sich nicht unbedingt um einen physischen Gegenstand handeln: Auch private Schlüssel und Zertifikate, die in der Kryptographie verwendet werden, können Nutzer authentifizieren. Die Gegenseite prüft, ob der Nutzer den richtigen privaten Schlüssel zu einem öffentlichen Schlüssel hat (siehe FIDO oder SQRL). Z.B. Browser können so die Echtheit von Webseiten überprüfen, wenn sie das Zertifikat der Seite übermittelt bekommen.

Der Vorteil der Besitz-basierten Authentifikation besteht darin, dass Nutzer sich keine Geheimnisse merken müssen, sondern lediglich den Besitz vorzuzeigen haben, um authentifiziert zu werden.

Nachteile dieser Art der Authentifikation sind, dass die benötigten Gegenstände verloren bzw. gestohlen werden können. Ohne sie kann der Nutzer sich nicht mehr authentisieren. Personen können sich zudem als andere ausgeben, wenn sie deren entsprechenden Besitz vorzeigen. Für bestimmte Objekte bedarf es besonderer Hardware, die Nutzer erst kaufen müssen, bevor sie Besitz-basierte Authentifizierung durchführen können (z.B. Kartenlesegerät für Chipkarten).

3.3 Authentifizieren durch Biometrie

Bei der Biometrie-basierten Authentifizierung werden Merkmale eines Menschen bzw. seines Körpers verwendet, um ihn als Nutzer zu erkennen. Biometrie kann in zwei Bereiche unterteilt werden: Der eine nutzt physische Merkmale, der andere Verhaltens-Merkmale.

Physische Merkmale sind statisch und ändern sich im Laufe der Zeit nicht oder nur ganz geringfügig. Hierzu zählen Fingerabdruck, Iris, Retina, Handgeometrie, Gesicht oder Handvenen. Neben statischen Merkmalen gibt es auch dynamische. Beispiele sind der Gang des Menschen, seine Unterschrift oder die Art des Anschlags, mit der er seine Tastatur bedient.

Nutzer können also nicht nur über körperliche Merkmale erkannt werden, sondern auch darüber, wie sie sich bewegen, was sie tun, wie sie sich verhalten. Jeder Nutzer hat auch einen charakteristischen Tages-Ablauf – wann er zum Beispiel zur Arbeit fährt, wie lange er arbeitet, wann er einkaufen geht usw. Auch das ist ein typisches Verhalten, an dem man Nutzer erkennen kann. Diese Eigenschaften lassen sich zur Identifizierung von Personen bzw. Menschen hinzuziehen. Für andere Entitäten wie Maschinen braucht es hingegen andere Eigenschaften.

Biometrie wird bisher in Webapplikationen so gut wie gar nicht bei der Authentifikation genutzt. Anders ist das zum Beispiel bei EasyPass[2], einem Grenzkontrollsystem. Dieses System wird auf den großen deutschen Flughäfen Frankfurt, Berlin, München und Hamburg eingesetzt. Nutzer gehen mit ihrem elektronischen Reisepass zu einem Scanner, schauen kurz in die Kamera und können dann die Grenzkontrolle passieren. Das System vergleicht das Kamerabild mit dem gespeicherten Bild auf dem Reisepass. Es muss kein Kontrolleur mehr präsent sein, der Nutzer kann den Vorgang alleine durchführen. Ein weiteres Beispiel sind Smartphones. Einige sind mit einem Fingerabdrucksensor ausgerüstet. Nutzer können dadurch ihr Telefon mit ihrem Fingerabdruck entsperren.

Biometrie hat den Vorteil, dass kein Gegenstand mitgeführt oder ein Geheimnis gemerkt werden muss. Nutzer können also nichts vergessen oder verlieren, was zu einer hohen Nutzerfreundlichkeit führt.

Es gibt allerdings auch einige Nachteile. Wenn ein Merkmal kompromittiert ist, also nicht mehr als vertrauenswürdig gilt, kann es nicht einfach ausgewechselt werden. Beispiel: Hat ein Angreifer einen Fingerabdruck gelesen und eine erfolgreiche Attrappe davon gebaut, kann der Nutzer den Fingerabdruck nur noch neunmal wechseln. Danach sind alle Möglichkeiten ausgeschöpft und das System kann nicht mehr sicher sein, ob es den Angreifer oder den Besitzer vor sich hat. Zudem ist der Vergleich der gespeicherten Daten mit den aktuell erfassten nie hundertprozentig gleich. Deshalb wird nur eine Wahrscheinlichkeit bzw. eine Ähnlichkeit der Daten bestimmt. Eine Kopie muss daher nur ähnlich genug sein, um als Original zu gelten Aus diesem Grund werden Lesegeräte für biometrische Merkmale entwickelt, die Verfahren nutzen, mit denen Fälschungs- und Lebenderkennung durchgeführt wer-

[2] `http://www.easypass.de`, besucht am 16.02.2017.

den kann. Ein weiterer Nachteil ist, dass solche speziellen und teuren Lesegeräte benötigt werden, um Biometrie nutzen zu können.

3.4 Kombinationen der Authentifikations-Methoden

Jede Methode aus den drei aufgeführten Kategorien kann gute Sicherheit bieten, wenn sie richtig implementiert und angewandt wird. Passwörter können sicher sein, wenn für jeden Account ein neues starkes Passwort verwendet wird. Oftmals ist das nicht der Fall. Zum Beispiel wird die Wahl des Passwortes oftmals dem Nutzer überlassen. Das ist für den Nutzer angenehm, führt aber, wie gezeigt, zu einer Verringerung der Sicherheit.

Um die Sicherheit einer Methode zu erhöhen, wird überlicherweise eine zweite hinzugefügt. Diese Kombination wird als *Zwei-Faktor-Authentifizierung, 2FA* bezeichnet, wenn die Methoden aus zwei verschiedenen Faktoren kommen. Die bekannteste Kombination ist die von Wissen mit Besitz.

Viele Webseiten bieten an, den passwortgeschützten Nutzeraccount mit einem zusätzlichen Passwort zu schützen. Dieses zweite Passwort ist dann bei jedem Authentifizierungsversuch anders (ein sogenanntes *one time password, OTP*) und wird mithilfe eines separaten Geräts generiert. Ein solches Gerät kann eine spezielle Hardware (z.B. RSA SecureID), ein USB-Token (z.B. Yubico U2F) oder ein Smartphone (z.B. mit Google Authenticator) sein. Dieser zusätzliche Code erhöht die Sicherheit auch solcher Accounts, die sonst nur durch ein schwaches Passwort geschützt sind. Dem Angreifer wird der Angriff auf ein Nutzerkonto erschwert, da er nun nicht nur das Passwort benötigt, sondern auch den Code, der sich jedesmal ändert. Dafür muss er sich in den Besitz der Hardware bringen und auf ihr den physischen Zugang herstellen. Immer mehr Webapplikationen bieten einen solchen Schutz für die Nutzer an, darunter auch Google, Dropbox und Github.

Das Kombinieren von Faktoren führt immer zu einer Authentifizierung mit mehreren Schritten. Zum Beispiel gibt der Nutzer erst sein Passwort ein und dann den Code der Hardware. Das reduziert die Benutzerfreundlichkeit. Benutzbarkeit und Sicherheit stehen sich oft umgekehrt proportional gegenüber (siehe Abbildung 3.2):

Je sicherer eine Authentifizierungsmethode ist, desto weniger nutzerfreundlich wird sie – Internetnutzer akzeptieren das nur schwerlich. Die Nutzer verlangen nach einer Mehr-Faktor-Authentifizierung, die sich aber nicht als eine solche anfühlt. Auch soll diese Authentifizierung nicht statisch, sondern dynamisch sein: Es soll zum Beispiel sofort erkannt werden, wenn ein Nutzer seinen Arbeitsplatz-Computer verlässt oder sein Smartphone aus der Hand legt. In diesem Fall soll der Zugang dann schnell geschützt sein.

Eine Lösungsmöglichkeit ist, dass Maschinen wie Computer, Smartphone, Smart Watch usw. ihre Besitzer anhand von Biometrie selbst erkennen [14]. Jeder Mensch ist einzigartig durch seine physischen Merkmale und durch sein Verhalten. Mithilfe verschiedener Sensoren in Hardware können diverse Merkmale erfasst und zu einer Darstellung eines Nutzers bzw. seines Profils zusammengefasst werden.

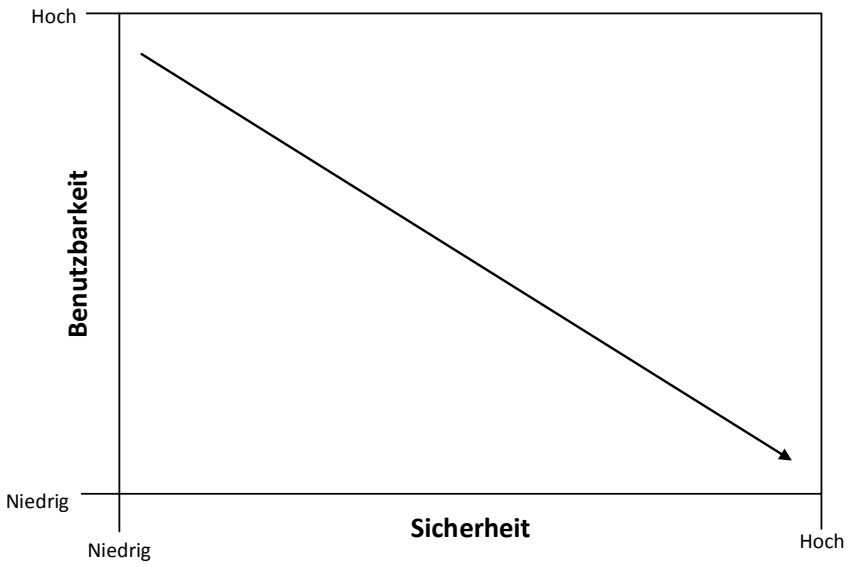

Abbildung 3.2: Verhältnis von Benutzbarkeit und Sicherheit

Es gibt viele Hardwarekomponenten, die relevante Daten erfassen können. Dazu zählen zum Beispiel Kamera, Mikrofon, Tastatur, Touchscreen, Beschleunigungssensor, GPS (Global Positioning System), EKG (Elektrokardiogramm), Fingerabdrucksensor usw. Durch regelmäßiges Auswerten der erfassten Daten kann eine kontinuierliche Authentifizierung ermöglicht werden.

Ein Nachteil der Nutzererkennung ist, dass es einige Zeit braucht, um ein Profil zu erstellen. Das Verhalten eines Nutzers variiert täglich, je nachdem ob es ein Arbeitstag ist, sich um ein Wochenende handelt oder ein Urlaub vorliegt. Sobald ausreichend Daten gesammelt sind, kann jedoch eine Authentifikation automatisiert im Hintergrund durchgeführt werden. Das steigert die Nutzerfreundlichkeit und durch die Analyse vieler Faktoren nimmt auch die Sicherheit zu. Eine beispielhafte Umsetzung wird in Abschnitt 5.2.2 über derzeitige Forschung im Hasso-Plattner-Institut beschrieben.

4 Technologien, Protokolle und Standards

Im Laufe der Jahre ist eine ganze Reihe von Technologien, Protokollen und Standards entwickelt worden, mit denen der zentralisierte, dezentralisierte oder föderierte Ansatz umgesetzt werden kann. Grundsätzlich gibt es dabei drei Rollen: den *Dienstanbieter* oder auch *Service Provider (SP)*, den *Identitätsprovider (IdP)* und den *Nutzer (User)*. Das Zusammenspiel der drei Rollen ist in Abbildung 4.1 dargestellt.

Der Identitätsprovider ist ein spezieller Dienstanbieter. Nutzer können dort ihre Daten wie Name, E-Mail-Adresse oder Wohnanschrift speichern und verwalten. Der Provider kann Dritten (z.B. andere Dienste) bestätigen, dass ein Nutzer eine bestimmte Identität hat. Nämlich die zu dem Nutzer gespeichert ist. Neben der Bestätigung kann er auch notwendige Attribute mit an den Dritten schicken. Der Nutzer muss dann keine Daten mehr eingeben, da sie ja auf dem IdP gespeichert sind.

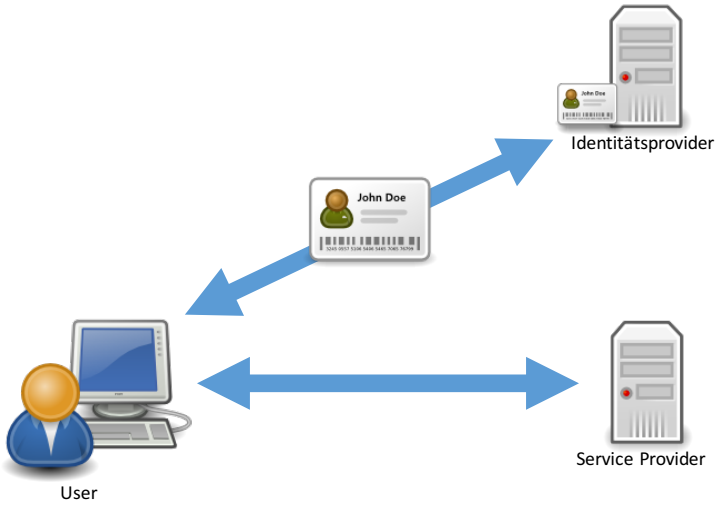

Abbildung 4.1: Zusammenspiel von Identitätsprovider, Service Provider und User

Bevor der User einen Dienst benutzen kann, muss er sich gegenüber dem Identitätsprovider authentisieren. Die möglichen Authentifikationsmethoden sind in

Kapitel 3 beschrieben. Nach der Authentifikation erhält der Nutzer ein sogenanntes Token, das die gewünschten Claims enthält, die der Service benötigt. Tokens werden häufig zur Authentifizierung eingesetzt. Sie enthalten Informationen, die für die Authentifikation wichtig sind (hier die Claims) und sind oft auch kryptografisch signiert oder verschlüsselt. Mit der Signatur lässt sich die Herkunft des Tokens überprüfen und die Verschlüsselung sorgt dafür, dass kein Dritter die Informationen des Tokens lesen kann. Chipkarten und USB-Sticks werden oft als Hardware-Tokens eingesetzt, aber auch virtuelle Software-Tokens sind möglich. Anhand der Signatur kann der Dienst überprüfen, ob das Token vom Identitätsprovider kommt und den Nutzer dann als authentifiziert ansehen. Hat ein Nutzer sich einmal über den Identitätsprovider bei einem Dienst angemeldet, kann eine Anmeldung bei einem zweiten Dienst automatisch erfolgen. Der Nutzer muss sich nicht erneut authentisieren. Diese Funktionalität wird als *Single-Sign-On (SSO)* bezeichnet.

Die nachfolgend aufgeführten Protokolle und Standards beschreiben, wie die Kommunikation zwischen den drei Rollen durchgeführt werden kann und welche Vor- und Nachteile es gibt. Ferner werden Anwendungsbeispiele dieser Protokolle aufgeführt. Vom Prinzip her sind alle Technologien sehr ähnlich, d. h. der Nutzer bekommt oftmals nicht mit, welche verwendet wird. Er sieht nur, dass er von seinem Dienst zu seinem Identitätsprovider umgeleitet und nach erfolgreicher Anmeldung wieder zum Dienst zurückgeleitet wird. Die Unterschiede sind meist technischer Natur.

4.1 Kerberos

Kerberos ist ein verteiltes Authentifizierungsprotokoll [20]. Der Identitätsprovider ist der Kerberos-Server (auch Key Distribution Center, KDC, genannt) und besteht aus zwei Komponenten: Authentifizierungsdienst (Authentication Service, AS) und Ticket-Granting Service (TGS).

Es werden Tickets bei der Authentifizierung verwendet. Ein Ticket ist eine andere Bezeichnung für Token. Es beweist die Identität eines Nutzers und ist kryptografisch verschlüsselt. Abbildung 4.2 veranschaulicht den Authentifikationsprozess. Um einen Dienst zu nutzen, muss zuerst ein Ticket-Granting Ticket (TGT) vom Authentifizierungsdienst angefordert werden. Dazu ist eine Anmeldung beim Server erforderlich, z. B. durch Eingabe eines Passwortes (1 + 2). Mit diesem TGT können Tickets für die jeweiligen Dienste bei dem Ticket-Granting Service angefordert werden (3 + 4). Mit einem Ticket kann der Dienst dann benutzt werden (5). Das Anfordern weiterer Tickets benötigt keine Re-Authentifizierung, solange das TGT gültig ist. Die Tickets enthalten die Sitzungsschlüssel (Session Keys), mit denen die eigentliche Kommunikation zwischen Server und User verschlüsselt wird. Der Austausch dieses Sitzungsschlüssels mit Kerberos basiert auf dem Needham-Schroeder-Protokoll [19] und funktioniert sowohl mit symmetrischer als auch asymmetrischer Verschlüsselung. Kerberos lässt sich unter anderem in Windows-basierten Netzwerken finden. Es ist das Standardprotokoll zur Authenti-

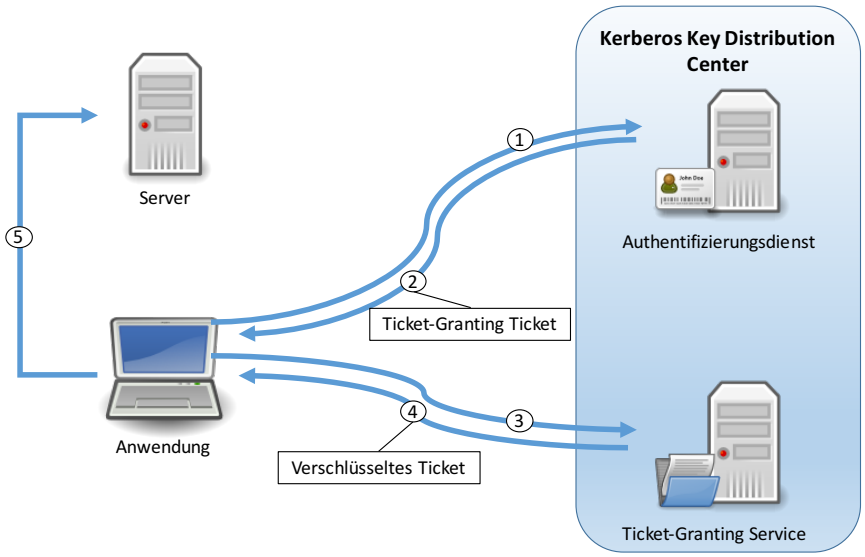

Abbildung 4.2: Kerberos – Workflow

fizierung ab dem Betriebssystem Windows 2000. Nach der Anmeldung an einem Netzwerk-Computer bekommt der Nutzer sein Ticket Granting Ticket. Wenn er damit Outlook öffnet, wird mit dem TGS ein Outlook Ticket erstellt. Outlook öffnet sich und kann ohne weitere Anmeldung verwendet werden.

Ein Vorteil von Kerberos ist das gegenseitige Authentifizieren von Client und Server. Das heißt, sowohl Server als auch Client wissen, dass ihr jeweiliger Partner der ist, der er vorgibt zu sein.

Ein Nachteil liegt im Einsatz eines zentrales Servers. Fällt dieser aus, ist eine Anmeldung nicht mehr möglich (Single Point of Failure). Wenn der Server kompromittiert ist, kann ein Angreifer jede Person imitieren. Kerberos hat auch strenge Zeitanforderungen, was eine Synchronisierung der Computer- bzw. Systemuhren aller Beteiligten voraussetzt.

4.2 Public-Key Infrastrukturen

Public-Key-Infrastrukturen (PKI) sind Systeme, die digitale Zertifikate ausstellen, verteilen und prüfen können. Zu einer solchen Infrastruktur gehören eine Zertifizierungsstelle (Certificate Authority, CA), eine Registrierungsstelle (Registration Authority, RA), ein Validierungsdienst (Validation Authority, VA) und ein Format für die Zertifikate. Die bekanntesten Infrastrukturen basieren auf dem X.509-Zertifikat [24].

Abbildung 4.3 zeigt, wie alle Komponenten zusammenspielen. Zuerst generiert ein Nutzer ein eigenes neues Schlüsselpaar für sich, bestehend aus einem öffentlichem und einem privaten Schlüssel (1). Mit dem öffentlichen Schlüssel wendet er

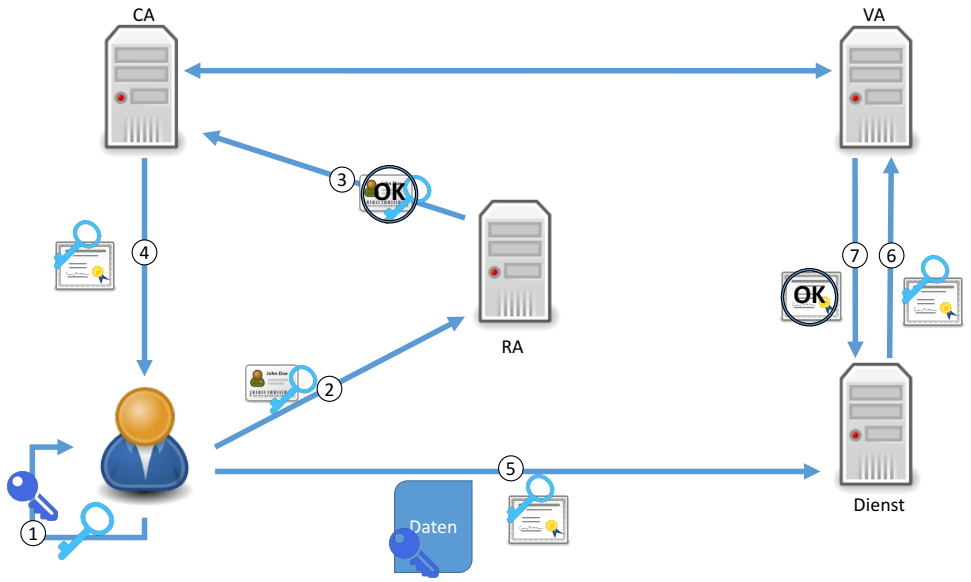

Abbildung 4.3: Public-Key-Infrastruktur – Workflow

sich an eine Registrierungsstelle (RA). Diese bestätigt die Echtheit des Schlüssels. Dabei wird geprüft, ob der Nutzer den passenden privaten Schlüssel hat (2). Ist alles korrekt, stellt die Zertifizierungsstelle (CA) das Zertifikat aus, das den öffentlichen Schlüssel enthält. Das Zertifikat der CA ist digital unterschrieben (3 und 4). Diese Schritte bilden den Registrierungsprozess. Der Nutzer kann beim Zugriff auf einen Dienst seine Daten mit seinem privaten Schlüssel unterschreiben. Die Daten zusammen mit dem Zertifikat werden zum Dienst übertragen (5). Der Dienst kann das Zertifikat über die Validierungsstelle (VA) auf Gültigkeit überprüfen lassen. Bei Gültigkeit des Zertifikats kann der Dienst auch die Daten als gültig ansehen, da sie mit dem passenden privaten Schlüssel signiert wurden (6 und 7). Das ist der Authentifikationsprozess.

Public-Key-Infrastrukturen kommen unter anderem bei TLS-Verschlüsselungen (Transport Layer Security, früher bekannt als Secure Socket Layer SSL, ist ein Verschlüsselungsprotokoll und wird vorwiegend bei HTTPS Verbindungen eingesetzt) oder in Chipkarten zum Einsatz. Es handelt sich dabei um streng hierarchische Infrastrukturen mit einer obersten Zertifizierungsstelle (der sogenannten Root CA), der alle vertrauen müssen. Da es schwierig ist, eine einzige weltweit einheitliche Instanz zu schaffen, gibt es viele Root CAs, z. B. bei TLS- Zertifikaten für Webbrowser (D-Trust, Telekom, Symantec usw.).

Neben diesen hierarchischen Modellen gibt es auch andere Ansätze. Der bekannteste ist das Web of Trust-Modell. Hier kann jeder ein Zertifikat unterschreiben. Ein Nutzer, der ein solches Zertifikat bekommt, muss selbst entscheiden, ob er dem, der unterschrieben hat, vertraut oder nicht. Umgesetzt ist das Web of Trust-Modell im OpenPGP-Standard [4] und wird unter anderem bei der Signierung und Verschlüsselung von E-Mails verwendet.

Public-Key-Infrastrukturen erlauben eine Authentifikation ohne vorherigen Kontakt. Es muss vorher kein Passwort oder dergleichen ausgetauscht bzw. vereinbart werden. Ein Dienst kann einen Nutzer authentifizieren, den er vorher noch nie gesehen hat. Es muss lediglich ein gültiges Zertifikat einer vertrauenswürdigen und akzeptierten Zertifizierungsstelle vorgelegt werden.

Dagegen bedarf es ein gutes Verständnis der PKI und der zugrunde liegenden asymmetrischen Kryptografie um sie richtig anzuwenden. Für einen Systemadministrator ist es nicht das Leichteste eine PKI einzurichten und auch für Endnutzer sind sie schwer verständlich.

4.3 WS-*

WS-* bezeichnet eine Reihe von Spezifikationen für Internetdienste (Web Services), basieren auf den SOAP- (Simple Object Access Protocol, [3]) und WSDL- (Web Services Description Language, [7]) Standards und erweitern diese.

Die relevanten Standards für Identitätsmanagement sind WS-Trust, WS- Federation, WS-SecurityPolicy und WS-MetadataExchange. WS-Trust beschreibt, wie man einen Security Token Service (STS) erzeugt. Dieser ist Teil des Identitätsproviders und generiert die Sicherheitstokens, welche die vom Dienst angeforderten Attribute enthalten. Der Standard erlaubt unterschiedliche Formate für die Tokens wie zum Beispiel X.509-Zertifikate, Kerberos-Tickets oder SAML-Assertions (Eine Assertion in SAML ist eine andere Bezeichnung für Claim bzw. Behauptung). Wie ein Token aufgebaut ist, wird im WS-Security Standard beschrieben. Mit WS-SecurityPolicy können Dienste spezifizieren, welche Attribute sie von Nutzern benötigen und welche der Identitätsprovider bereitstellen sollte. Auf der anderen Seite kann der IdP mit WS-MetadataExchange angeben, welche Claims er verwaltet und zur Verfügung stellen kann. WS-Federation ist eine Erweiterung von WS-Trust und erlaubt die Implementierung des föderierten IdM-Ansatzes. Microsofts ADFS (Active Directory Federation Services) implementiert diese Standards.

Die WS-*-Spezifikationen sind auch dafür entwickelt worden, um ein Identitäts-Metasystem aufbauen zu können. Bestehende IdM-Systeme, die unterschiedliche Tokenformate nutzen, mussten so nicht geändert werden. Es mussten nur Komponenten wie STS, SecurityPolicy usw. in die bestehenden Systeme integriert werden, um so eine Föderation aufbauen zu können.

Es ist ein Vorteil dieser Spezifikationen, dass Systeme nicht geändert, sondern lediglich neue Komponenten integriert werden müssen. Durch das zugrunde liegende SOAP-Protokoll können auch Dienste miteinander verbunden werden, die andere Transportprotokolle verwenden, zum Beispiel SMTP (Simple Mail Transfer Protocol) statt HTTP (HyperText Transfer Protocol, Standard Protokoll zur Datenübertragung im Internet).

Die zu übertragenden Daten werden in WS-* durch die SOAP-Basis als XML repräsentiert. XML-Dateien können schnell sehr komplex und unverständlich werden.

4.4 WebID

WebID bietet die Möglichkeit, eine Person, Organisation oder andere Entitäten eindeutig zu identifizieren. Dazu wird eine URI (Uniform Resource Identifier, einheitlicher Bezeichner von Ressourcen; URLs sind spezielle URIs) benutzt, die auf ein RDF-Dokument (Resource Description Framework) verweist [23]. Das Dokument enthält die Attribute des Nutzers und kann mit RDF- Vokabular beschrieben werden. Ein Beispiel für ein Vokabular ist FOAF (Friend Of A Friend).[1] Der Server, auf dem das Dokument liegt, ist der Identitätsprovider.

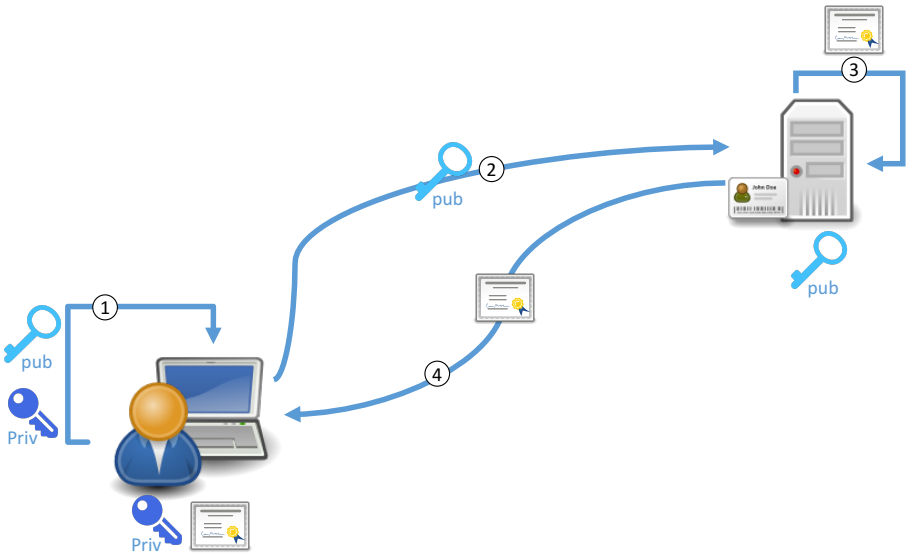

Abbildung 4.4: WebID – Erstellung eines Zertifikates

Um sich bei Diensten mit der WebID anzumelden, muss der Nutzer beweisen, dass ihm die URI bzw. das Dokument auf dem IdP gehört. Dazu werden X.509-Zertifikate eingesetzt. Um ein Zertifikat zu erstellen, bietet der IdP eine entsprechende Operation an. Der Browser generiert lokal ein asymmetrisches Schlüsselpaar (1) und sendet den öffentlichen Teil an den IdP (2). Dort wird ein Zertifikat mit der URI erstellt, vom IdP signiert (3) und dann wieder an den Browser gesendet (4). Der Browser speichert den privaten Schlüssel und das Zertifikat, und der IdP speichert den öffentlichen Schlüssel im RDF-Dokument. Abbildung 4.4 zeigt den Ablauf in grafischer Darstellung.

Die Anmeldung mit WebID bei Diensten zeigt Abbildung 4.5. Ist die Anmeldung mit WebID erlaubt, wählt der Nutzer die Identität bzw. das Zertifikat, welche er

[1] `http://www.foaf-project.org`, besucht am 16.02.2017.

benutzen will, aus und sendet dies dann dem Dienst (1). Dort kann mittels der URI auf den Identitätsprovider zugegriffen und das RDF-Dokument gelesen werden (2 + 3). Dann vergleicht der Dienst die öffentlichen Schlüssel aus dem RDF-Dokument mit dem Schüssel aus dem Zertifikat (4). Bei Übereinstimmung wird dem Nutzer Zugriff ermöglicht (5).

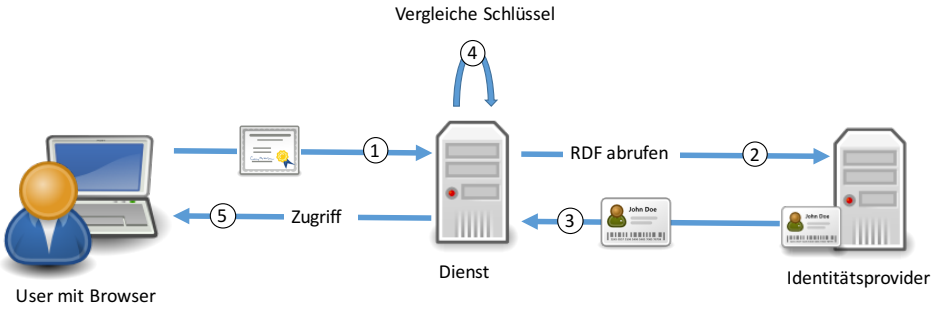

Abbildung 4.5: WebID – Authentifizierung mit WebID

Ein Vorteil von WebID ist, dass durch die Verwendung von Zertifikaten kein Passwort nötig ist. Zertifikate werden im Browser gespeichert, deshalb kann auf Browser-Funktionen zurückgegriffen werden, um sich so besser gegen Angriffe, z.B. Phishing, zu schützen.

Ein Nachteil von WebID ist, dass das Profil immer erreichbar sein muss, um das Zertifikat zu prüfen. Bekommt außerdem jemand Zugriff auf den Browser, kann er sich unter anderem Namen bei Diensten anmelden.

4.5 Mozilla Persona (BrowserID)

Mozillas Persona (oder auch bekannt als BrowserID) ist WebID sehr ähnlich. Die BrowserID bzw. die genutzte ID ist hier keine URI, sondern die E-Mail-Adresse. Die Idee kommt daher, dass viele Nutzer bereits verstehen, dass eine E-Mail einen Nutzer eindeutig identifiziert und sich somit gut als Identifier eignet. Dadurch, dass viele Nutzer mehrere E-Mail-Adressen für unterschiedliche Zwecke haben, können auch unterschiedliche Identitäten für unterschiedliche Dienste genutzt werden. Ähnlich wie bei WebID werden Zertifikate verwendet. Ein Unterschied zu WebID

ist, dass keine X.509-Zertifikate benutzt werden, sondern Zertifikate auf JSON-Basis (JavaScript Object Notation ist ein einfach lesbares und kompaktes Datenformat). Die JSON-Zertifikate basieren auf dem JWT (JSON Web Token) Standard. Die Informationen in solch einem Token sind digital signiert. Das Signieren und auch Verschlüsseln kann z.B. mit dem JOSE (Javascript Object Signing and Encryption) Framework bewerkstelligt werden. Der Ablauf zum Erstellen eines Zertifikates ist in Abbildung 4.6 dargestellt.

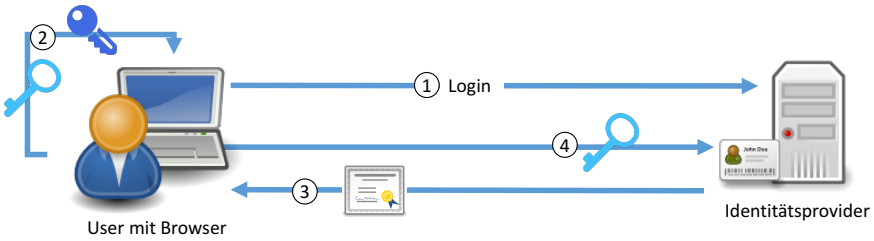

Abbildung 4.6: BrowserID – Erstellung von Zertifikaten

Zunächst meldet sich der Nutzer bei einem BrowserID-Identitätsprovider an (z.B. mit E-Mail und Passwort) (1). Dort muss der Nutzer beweisen, dass er die E-Mail-Adresse besitzt, z.B. durch eine Bestätigungs-E-Mail. Das ist nur nötig, wenn der IdP und der E-Mailprovider verschieden sind. Danach wird über den Browser ein Schlüsselpaar generiert (wie bei WebID) (2). Der öffentliche Schlüssel wird an den IdP gesendet (3), der darauf hin das Zertifikat für den Nutzer erstellt (4). Das Zertifikat enthält die E-Mail-Adresse, den öffentlichen Schlüssel und einen Gültigkeitszeitraum. Dieses Zertifikat wird vom IdP signiert und an den Nutzer geschickt (5), wo es im Browser zusammen mit dem privaten Schlüssel gespeichert werden kann.

Zum Anmelden bei einem Dienst (Abbildung 4.7), der BrowserID unterstützt, wählt der Nutzer seine ID aus. Der Browser generiert dann eine Zusicherung (Assertion), welche das Zertifikat und die URL des Dienstes enthält und signiert die Zusicherung mit dem privaten Schlüssel des Nutzers (1). Diese Zusicherung wird zusammen mit dem Zertifikat zum Dienst übertragen (2). Der Service lädt sich den öffentlichen Schlüssel des Identitätsproviders herunter (3). Die Domäne

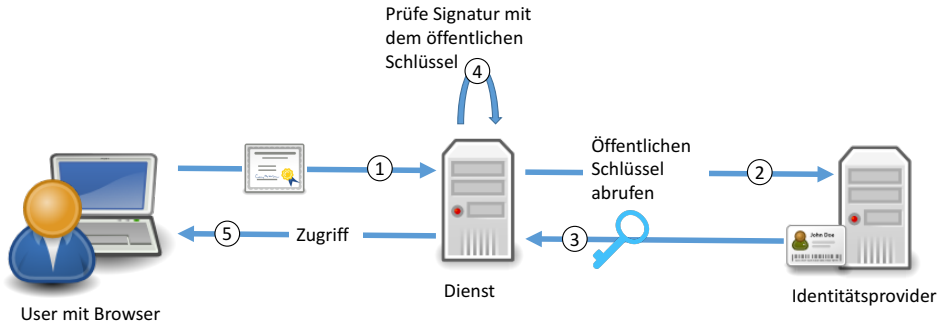

Abbildung 4.7: BrowserID – Validation von Zertifikaten

des IdP ist in dem Zertifikat angegeben. Damit können das Zertifikat und der öffentliche Schlüssel des Nutzers überprüft werden, womit dann anschließend auch die Zusicherung verifiziert werden kann (4).

Ein Vorteil von BrowserID ist die Nutzung der E-Mail-Adresse als eindeutige ID. Da die meisten Nutzer bereits ein gutes Verständnis von E-Mail-Adressen haben, ermöglicht es eine einfachere Anwendung im Vergleich zu einer URI wie bei WebID. Dadurch, dass vielen Diensten ohnehin die E-Mail-Adresse vorliegt, kann BrowserID direkt Verwendung finden. Ein weiterer Vorteil von BrowserID gegenüber den meisten anderen Technologien ist, dass der Identitätsprovider nicht weiß, welcher Nutzer welchen Dienst verwendet. Der IdP weiß nur, welche Dienste den öffentlichen Schlüssel des Dienstes anfordern, aber nicht, welcher Nutzer damit validiert wird. Das schützt die Privatsphäre der Nutzer.

Ein Nachteil von BrowserID ist, dass es noch keine native (standardmäßige) Einbindung in den Browser gibt, um die Zertifikate zu speichern, wie es in WebID der Fall ist. Außerdem wird BrowserID nicht mehr aktiv weiterentwickelt.

4.6 OpenID

OpenID [13] ist ein Authentifizierungsprotokoll und nutzt eine URL zum Identifizieren von Nutzern. Das ist ähnlich zu WebID, wo eine URI verwendet wird. Anders als bei WebID verweist die URL nicht auf ein Dokument mit dem Nutzerprofil, sondern auf ein Dokument, das die Adresse des IdP enthält. Dadurch braucht der Dienst keine OpenID-Provider im Vorfeld zu kennen, sondern kann die URL-Adresse über die OpenID auslesen, wenn sich ein Nutzer anmeldet. Das ent-

spricht dem dezentralen Identitätsmanagement-Ansatz. Im Gegensatz zu WebID wird die OpenID als Benutzername verwendet statt einem Zertifikat. Der Ablauf des Protokolls ist in Abbildung 4.8 dargestellt.

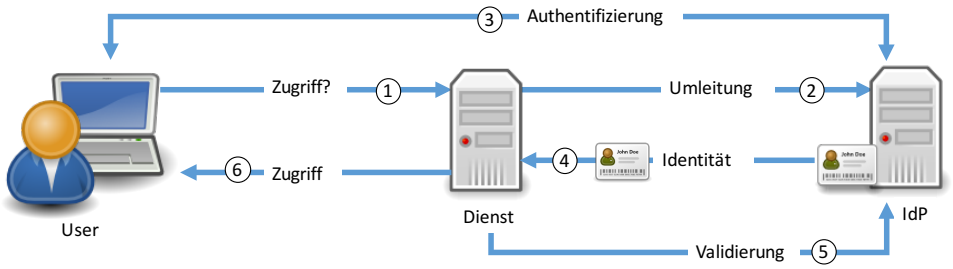

Abbildung 4.8: OpenID – Workflow

Zum Anmelden bei einem Dienst gibt der Nutzer seine OpenID in einem Eingabefeld ein. Der Dienst ermittelt über diese ID den dazugehörigen Identitätsprovider und leitet den Nutzer dorthin um (1 + 2). Auf der Seite seines IdP meldet sich der Nutzer an (3). Der IdP stellt dann das Identitätstoken zusammen und leitet den Nutzer, zusammen mit dem Token, zurück zum Dienst (4). Nach Validierung des Tokens (5) kann der Nutzer den Dienst verwenden (6). Für die Verifikation des Tokens sind zwei Möglichkeiten spezifiziert. Zum Einen kann der Dienst nach Erhalt des Tokens die Daten – unabhängig vom Nutzer – an den Provider zur Überprüfung senden. Zum Anderen kann zu Beginn der Kommunikation zwischen Dienst und Provider ein Geheimnis ausgetauscht werden. Mit diesem Geheimnis kann ein Message Authentication Code (MAC) als Signatur berechnet werden, die auf der Seite des Dienstes verifiziert werden kann. OpenID wird unter anderem bei Stackoverflow[2] eingesetzt, einem Portal, auf dem Entwickler Fragen und Antworten austauschen. OpenID wurde früher bei reichweitenstarken Webseiten wie Google, Facebook usw. verwendet. Durch das Eingeben einer URL als Benutzernamen wurde OpenID nicht so viel verwendet, obwohl viele Nutzer eine OpenID durch ihren E-Mail-Provider hatten. Das Merken und Eingeben der URL ist nich so

[2] `http://stackoverflow.com`, besucht am 16.02.2017.

benutzerfreundlich wie ein einfacher Name. Zusätzlich kommen noch einige kleine Sicherheitsaspekte dazu. So wurde OpenID nach und nach durch OAuth ersetzt (siehe nächsten Abschnitt). Daher lässt sich OpenID nur noch auf vereinzelten Seiten finden, wie eben Stackoverflow.

Der Vorteil von OpenID ist, dass Dienste im Vorfeld keine Provider kennen müssen, da diese über die ID ermittelt werden können. Auch kann jedermann einen OpenID-Provider schaffen und betreiben.

Darin liegt aber auch der Nachteil von OpenID: Wenn jedermann einen Provider betreiben kann, sind die Identitäten, die ausgestellt werden, nicht in jedem Fall vertrauenswürdig. OpenID ist auch anfällig gegen Angriffe wie zum Beispiel Phishing (siehe Sektion 2).

4.7 OAuth

OAuth [17] steht für Open Authorization und ist im Gegensatz zu OpenID ein Protokoll zur Autorisierung (siehe Abschnitt 1.2), nicht zur Authentifizierung. Es erlaubt Nutzern, anderen Personen Zugriff auf ihre Dokumente oder andere Ressourcen zu geben, ohne die eigenen Anmeldeinformationen weiterzugeben. Es kann dann für die Authentifikation benutzt werden, wenn die autorisierten Dokumente Informationen über die Identität enthalten. Oftmals spricht man von einer Pseudo-Authentifikation. Im OAuth-Vokabular ist der Nutzer ein Resource Owner (RO) und der Identitätsprovider kann aufgeteilt werden in Ressource Server (RS) und Authorization Server (AS). Der AS übernimmt die Authentifikation. Die eigentlichen Daten liegen auf dem RS. Es kann sich auch um denselben Server handeln. Der Dienst, bei dem sich der Nutzer anmelden will und der die Identität benötigt, wird auch als Relying Party (RP) bezeichnet.

Den Prozess der Authentifikation mit OAuth zeigt Abbildung 4.9. Damit sich der Nutzer bei einem Dienst anmelden kann, wählt er dort einen Identitätsprovider aus, auf den hingewiesen wird. Daraufhin erfolgt eine Weiterleitung zum IdP (1 + 2). Wie bei OpenID meldet sich der Nutzer an. Zusätzlich muss er eine Zustimmung (Consent) erteilen, dass der Dienst auf die Identitätsdaten des Nutzers zugreifen darf (3). Danach erhält der Dienst einen Autorisierungscode (Authorization Code) (4). Dieser Code kann beim IdP gegen ein Zugriffstoken (Access Token) eingetauscht werden (5). Der Nutzer ist hier nicht mehr direkt involviert. Mit dem Zugriffstoken kann der Dienst nun auf die autorisierten Identitätsdaten zugreifen (6). Danach kann der Nutzer Zugriff auf den Dienst erhalten. Der Autorisierungscode ist also eine Genehmigung des Nutzers an den Dienst, mit dem er sich einen Schlüssel (Access Token) für die beantragten Attribute ausgeben lassen kann. So lange der Schlüssel gültig ist, kann der Dienst dann jederzeit auf die Daten zugreifen. OAuth wird auf den reichweitenstarken Webseiten wie Google, Paypal, Twitter usw. eingesetzt.

Die Vorteile von OAuth liegen in dessen Einfachheit für Entwickler und in der großen Verbreitung.

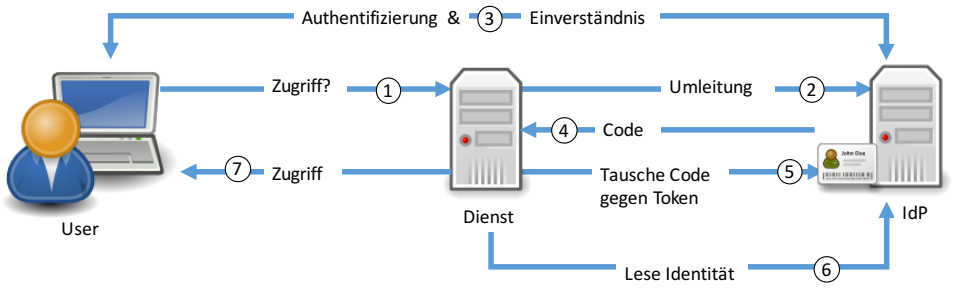

Abbildung 4.9: OAuth – Workflow

Nachteil von OAuth sind fehlende Sicherheitsfeatures im Protokoll. Es wird keine Verschlüsselung oder Signatur vorgeschrieben, sondern es wird darauf vertraut, dass TLS (Transport Layer Security) verwendet wird. Auch wissen Dienste nur, dass ein Zugriffstoken autorisiert wurde, aber nicht durch welchen Nutzer. Diese Unsicherheit macht OAuth zu einem Pseudo-Authentifikationsprotokoll.

4.8 OpenID Connect

OpenID Connect (OIDC) [22] ist der Nachfolger von OpenID, basiert aber auf dem OAuth-Protokoll. OpenID Connect fügt OAuth eine einfache Identitätsschicht hinzu. Damit wird aus OAuth ein echtes Authentifikationsprotokoll.

Abbildung 4.10 zeigt, worin der Unterschied zwischen OAuth und OpenID Connect liegt. Beim Tausch des Autorisierungscodes gegen das Zugriffstoken erhält der Dienst zusätzlich ein Identitätstoken (ID Token) (5). Dieses Token enthält einen Identifier des autorisierenden Nutzers. Zusätzlich beinhaltet das Token alle Informationen zur Verifikation: über den IdP, der das Token ausgestellt hat, das Datum, die Gültigkeitsdauer, den Dienst, für den das Token bestimmt ist und noch weitere Attribute wie z.B. die Authentifizierungsmethode. Das ID Token wird vom Identitätsprovider digital signiert und kann bei Bedarf auch verschlüsselt werden. Wenn der Dienst also seinen Schlüssel (Access Token) abholt, bekommt er zusätzlich vom Identitätsprovider ein Dokument, das Informationen über denjenigen, der die Genehmigung (Authorization Code) ausgestellt hat. Wenn die Identitätsdaten mit dem Zugriffstoken abgerufen werden (6), kann der Dienst die Nutzeridentifier aus dem ID Token und den Identitätsdaten vergleichen und ist dann sicher, ob

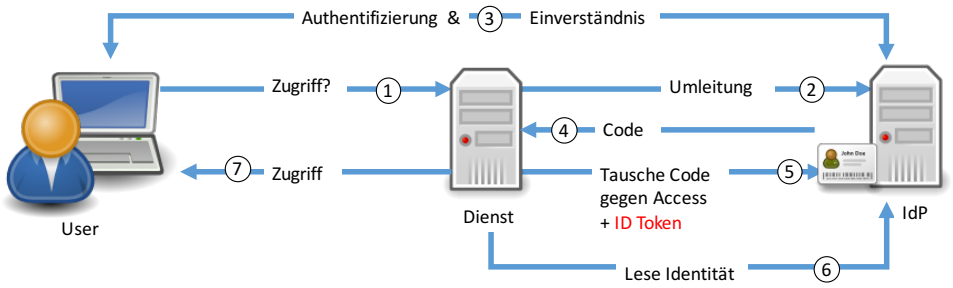

Abbildung 4.10: OpenID Connect – Unterschied zu OAuth

der Zugriff durch die richtige Person autorisiert wurde oder nicht. Anders als bei OpenID und OAuth spezifiziert OIDC ein Format für das Identitätstoken. Das verwendete Token-Format ist JWT (JSON Web Token). Mithilfe des JOSE- (Json Object Signing and Encryption) Frameworks wird ein JWT verschlüsselt und signiert. Dienste wie Google oder Paypal haben bereits den OpenID Connect-Standard implementiert.

Vorteil von OIDC ist, dass es ebenso einfach zu verwenden ist wie OAuth und man kann sehr leicht OAuth zu OpenID Connect erweitern. Auch unterstützt OIDC native und mobile Applikationen.

Als Nachteil ist zu nennen, dass OpenID Connect nicht kompatibel ist mit OpenID 2.0. Ein Nutzer kann seine OpenID nicht bei einem OpenID Connect System verwenden und die OIDC Connect Authentifizierung funktioniert nicht bei OpenID Systemen.

4.9 UMA

UMA (User Managed Access) ist, wie OpenID Connect, eine Erweiterung (Profil) von OAuth 2.0 [16]. UMA soll dem Nutzer mehr Kontrolle über seine Daten und den Zugriff durch Dritte geben. Es werden drei Hauptkonzepte hinzugefügt. Das erste Konzept definiert eine standardisierte API (Application Programming Interface, deutsch: Programmierschnittstelle) des Autorisierungsservers. Eine API oder auch Programmierschnittstelle gibt eine Reihe von Funktionen vor, die von anderen Komponenten oder auch anderen Systemen eingebunden werden können. Wie im Zusammenhang mit OAuth erwähnt, kann der Autorisierungsserver (AS) von

den eigentlichen Daten getrennt werden. Gibt es mehrere dieser Ressourcen-Server, müssen alle mit dem AS kommunizieren. Diese Kommunikation ist in OAuth nicht spezifiziert. Damit die Kommunikation einheitlich und übersichtlich bleibt, hat UMA eine API definiert. Die Kommunikation ist mit OAuth-Zugriffstoken gesichert. Es ist so sehr leicht, weitere Ressourcenserver hinzuzufügen. Das zweite Konzept beschreibt eine formale Notation von Diensten. Damit können Nutzer in Policies beschreiben, was ein Dienst tun darf. Zum Beispiel kann einem Dienst erlaubt werden, Zugriffsrechte auf Nutzerdaten an andere Dienste zu delegieren, ohne dass der Nutzer das erneut autorisieren oder seine Zustimmung geben muss. Das funktioniert so lange wie die Policy gültig ist. Das dritte große Konzept besagt, dass ein Dienst Zugriff auf Ressourcen bekommen kann, ohne dass der Nutzer aktiv zustimmen muss. Das geht aber nur, wenn der Dienst genug Behauptungen (Claims) aufweisen kann, die der Nutzer in einer Policy festgelegt hat und so für den Nutzer vertrauenswürdig ist.

4.10 SAML

SAML steht für Security Assertion Markup Language [8] und ist ein Authentifikationsprotokoll. Der Workflow von SAML ist etwas einfacher als bei OpenID und OAuth. Abbildung 4.11 stellt den Ablauf dar. Auch hier wird der Ablauf durch den

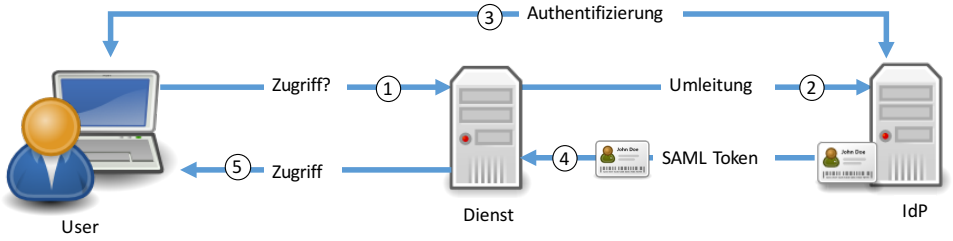

Abbildung 4.11: SAML – Workflow

Nutzer initiiert, wenn er auf einen Dienst zugreifen will (1). Dann erfolgt eine Weiterleitung zum Identitätsprovider (2), bei dem sich der Nutzer anmelden muss (3).

Es wird ein SAML-Token ausgestellt und an den Dienst geschickt (4). Der Nutzer hat nun Zugriff auf den Dienst (5). SAML basiert auf XML. Deshalb ist das Token ein XML-Dokument. Ein SAML-Token besteht aus den Claims des Nutzers, die SAML Assertions heißen. Ein Token kann über die Spezifikationen XML Signature und XML Encryption verschlüsselt und signiert werden. SAML wird vorwiegend in Unternehmens-Umgebungen eingesetzt.

Vorteil des Protokolls ist, dass es mit verschiedenen Transportprotokollen verwendet werden kann. Während OpenID (Connect) und OAuth auf HTTP basieren, kann SAML auch mit anderen Protokollen wie SMTP (Simple Mail Transfer Protocol) übertragen werden.

SAML ist mit XML deutlich komplexer als OpenID Connect oder OAuth, was es für Entwickler weniger attraktiv macht. Das ist ein Nachteil. Auch wurde das Protokoll anfänglich nur für Webapplikationen entwickelt und nicht für native Applikationen.

4.11 SCIM

SCIM steht für System for Cross-Domain Identity Management (früher auch bekannt als Simple Cloud Identity Management) [15]. SCIM ist kein Authentifikationsprotokoll, sondern definiert ein Datenschema für Ressourcen wie User oder Gruppen. Es wurde entwickelt, um die Verwaltung von Identitäten in Cloud-basierten Applikationen und Diensten zu vereinfachen. Ohne ein einheitliches Datenschema wird jeder Dienst seine eigene Struktur für Identitätsdaten verwenden. Wenn dann eine neue Koorperation eingegangen wird, müssen diese Daten in das Format des Partners umgewandelt werden. Mit einem standardisierten Schema entfällt die Umwandlung und der damit verbundende Arbeitsaufwand. Es werden allgemein gebräuchliche Attribute vordefiniert. Darunter ID, Benutzername, E-Mail oder auch die Adresse. Mit der Definition einer Gruppe können Organisationsstrukturen modelliert werden. Gruppen enthalten Nutzer oder andere Gruppen. SCIM definiert nur einfache Gruppenattribute. Eine genauere Spezifikation von Gruppen und Rollen bringt die Erweiterung VOOT. Neben der Definition des User- und Gruppen-Schemas, spezifiert SCIM auch eine REST[11]-API. Mit Hilfe der API können zwei Dienste einfach Daten austauschen. Die API gibt zum Beispiel vor, mit welchen *Befehlen (URLs)* Nutzeridentitäten erstellt, gelesen oder aktualisiert werden. Damit die URL nicht von jedermann benutzt werden, können sie mit OAuth bzw. OpenID Connect gesichert werden. Nur wer die benötigten Rechte bzw. Tokens hat, darf Identitäten verändern. Als Beispiel für die Anwendung von SCIM kann der Eintritt eines neuen Mitarbeiters in ein Unternehmen herangezogen werden. Wird der Nutzeraccount von den Administratoren erstellt, kann mithilfe der API bei allen Services, die der Mitarbeiter braucht, ebenfalls ein Account automatisch erstellt und konfiguriert werden. Verlässt der Mitarbeiter das Unternehmen, werden mit Löschung des Mitarbeiteraccounts automatisch alle verknüpften Informationen anderer Dienste gelöscht. Die beiden Vorgänge werden als Provisioning und De-Provisioning bezeichnet.

Mit SCIM können Daten bereitgestellt (Provisioning) werden. Wird bei einem Dienst oder IdP ein Nutzer angelegt, kann er auch bei allen anderen Diensten, welche die SCIM API implementieren, erstellt werden. Wenn ein Nutzer vom IdP gelöscht wird, dann kann er auch von allen anderen Diensten automatisch gelöscht werden. Diese Automatisierung ist ein Vorteil von SCIM. Es gibt eine Erweiterung von SCIM – VOOT genannt –, die sich detaillierter mit der Gruppen- und Rollenspezifikation befasst.

Der Nachteil von SCIM ist, dass es nur zum Erstellen und Lesen von Identitätsdaten dient. Bei Benutzung der API-Funktionen wird davon ausgegangen, dass der Ausführende berechtigt ist (z.B. durch ein gültiges Token). Wie die Berechtigung erlangt wird, muss mit anderen Mitteln umgesetzt werden, zum Beispiel mit OpenID Connect oder OAuth.

4.12 SQRL

SQRL[3] steht für Secure, Quick, Reliable Login (Sicherer, schneller, zuverlässiger Login; früher auch als Secure QR Login bekannt). Es ist ein offener Standard für die Nutzerauthentifikation. Das Besondere an diesem Konzept ist, dass es ohne dritte Entität, also ohne Identitätsprovider, auskommt. SQRL verspricht eine anonyme Authentifikation, bei der keine Informationen über den Nutzer an den jeweiligen Dienst preisgegeben werden. Das Einzige, was geteilt wird, ist ein öffentlicher Schlüssel. Der Schlüssel dient dabei als Identifier beim Dienst und kann auch als SQRL ID bezeichnet werden. Abbildung 4.12 zeigt den Ablauf einer Anmeldung mit SQRL. Will der Nutzer sich bei einem Dienst anmelden, wird dort ein QR-Code angezeigt (1). Der Code kann z.B. mit dem Smartphone gescannt werden (2). Der QR-Code enthält die URL des Dienstes und eine kryptografische Challenge bzw. Nonce (used only once[19]) – eine nur einmal verwendete zufällige Zeichenkette. Aus der URL und einem auf dem Smartphone befindlichen Masterschlüssel wird ein Schlüsselpaar (privater und öffentlicher Schlüssel) für den Dienst abgeleitet (3). Der Masterschlüssel wird bei der Installation der App einmalig generiert. Mit dem privaten Schlüssel wird die Nonce signiert und wird zusammen mit dem öffentlichen Schlüssel zum Dienst gesendet (4). Ist die signierte Nonce korrekt, bekommt der Nutzer Zugriff (5). Durch die Kombination von Masterschlüssel und URL entsteht für jeden Dienst ein neues Schlüsselpaar. Die einzelnen Schlüsselpaare werden so erzeugt, dass es nicht möglich ist zu unterscheiden, ob ein Schlüsselpaar von diesem oder einem anderen Smartphone generiert wurde. Dadurch ist es zwei Diensten nicht möglich, gemeinsame Nutzer zu erkennen. Da alle Schlüsselpaare von dem Masterschüssel abhängen, ist dieser mit einem Masterpasswort geschützt.

Der Vorteil von SQRL ist die Anonymität, welche die Nutzer haben, da nur der öffentliche Schlüssel übertragen wird. Neben dem Browser bedarf es auch eines Smartphones oder eines anderen Geräts, das den Masterschlüssel bzw. die Schlüs-

[3] `https://www.grc.com/sqrl/sqrl.htm`, besucht am 16.02.2017.

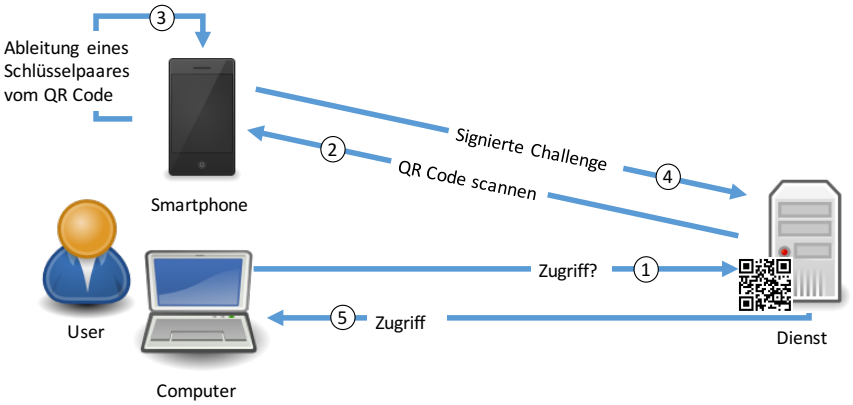

Abbildung 4.12: SQRL – Authentifizierung durch Scannen von QR-Codes

selpaare enthält. Das Gerät ist mit einem Masterpasswort geschützt. Durch diese zwei Faktoren (Gerät und Passwort) erhöht sich die Sicherheit der Authentifikation.

Ein Nachteil von SQRL ist, dass Nutzer sich um ihre Identität selbst kümmern müssen. Sie müssen den Diebstahl des Gerätes oder des Masterpasswortes verhindern und sich im Falle von Datenverlust selbst um Backups kümmern. Auch benötigen die meisten Dienste Informationen über den Nutzer, wodurch die Anonymität wieder aufgehoben wird. Identitätsdaten, die vom Nutzer kommen, sind für den Dienst nicht immer vertrauenswürdig. Ein ähnliches Verfahren lässt sich auch in BitCoin-Applikationen finden. Dort heißt es BitID.[4]

4.13 FIDO

Die FIDO (Fast IDentity Online) Allianz[5] definiert zwei Standards U2F und UAF. U2F steht für Universal 2nd Factor. Dieser Standard beschreibt eine allgemeine Zwei-Faktor-Authentifizierung. Damit können Passwort-basierte Web- Authentifikationssysteme mit einem zusätzlichen Faktor abgesichert werden. Ein zusätzlicher Faktor kann z.B. ein USB-Stick sein. Google, Github oder Dropbox unterstützen bereits das U2F-Protokoll. Mit Hilfe eines solchen U2F unterstützenden Gerätes wird die Sicherheit der Nutzerkonten erhöht. Der zweite Standard ist UAF (Univer-

[4] `https://github.com/bitid/bitid/blob/master/BIP_draft.md`, besucht am 16.02.2017.
[5] `https://fidoalliance.org`, besucht am 16.02.2017.

sal Authentication Framework). Dieser steht für eine allgemeine Authentifikation. Diese kann mit unterschiedlichen Faktoren durchgeführt werden, darunter auch biometrische Verfahren wie Fingerabdruck, Gesichts- oder Iriserkennung. Das Besondere daran ist: Der Nutzer authentifiziert sich gegenüber einem lokalen Gerät (FIDO Authenticator, z. B. ein Smartphone). Wie bei SQRL ist keine dritte Partei involviert. Das Gerät kommuniziert direkt mit dem Dienst, bei dem sich der Nutzer anmelden will. Auch hier wird auf der Webseite des Dienstes nur ein öffentlicher Schlüssel benötigt (meistens mit Nutzernamen zum Ansprechen des Nutzers). Die Registrierung mit FIDO ist in Abbildung 4.13 dargestellt. Das Authentifizieren funktioniert genau wie die Registrierung, nur mit einem kleinen Unterschied. Wenn

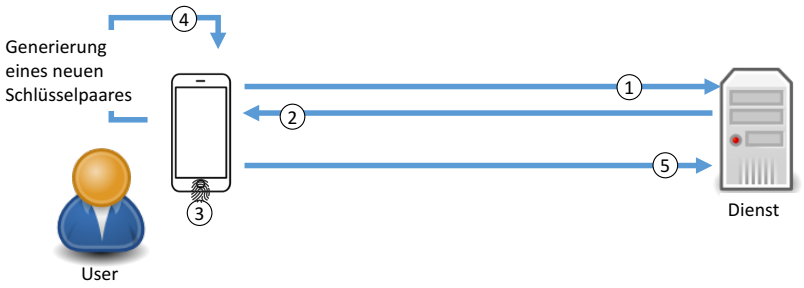

Abbildung 4.13: FIDO – UAF Registrierungsprozess

der Nutzer sich für einen Dienst registrieren will (1), bekommt er vom Dienst eine Aufforderung, sich auszuweisen (2). Mit dieser Aufforderung zusammen wird eine Challenge (zu deutsch: Herausforderung) gestellt, die der Nutzer erfüllen muss. Er nutzt sein Smartphone als „Authenticator“, um seine Identität zu bestätigen. Das kann er zum Beispiel mit seinem Fingerabdruck machen (3). Das Smartphone wird dann ein neuen privaten und öffentlichen Schlüssel generieren (4). Mit dem privaten Schlüssel wird die in (2) erhaltenen Challenge signiert. Die Signatur wird zusammen mit dem öffentlichen Schlüssel an den Dienst gesendet, um die Registrierung abzuschließen (5). Will der Nutzer sich nach einer Registrierung authentifizieren, wird bei Schritt (4) kein neues Schlüsselpaar erzeugt, sondern der entsprechende private Schlüssel ausgewählt. Dann wird wie bei der Registrierung eine Challenge signiert und das Ergebnis an den Dienst gesendet, um die Anmeldung abzuschließen.

Der Vorteil von FIDO ist, dass es den Nutzern unterschiedliche Authentifizierungsvarianten ermöglicht. Je nach Vorliebe der Nutzer, des Angebots des FIDO Authenticators oder der Anforderung des Dienstes, kann mal die eine, mal die andere Methode gewählt werden. Es können auch zwei Methoden nach U2F kombiniert werden. Eine Attacke auf den Dienst bringt dem Angreifer nichts, da er mit dem öffentlichen Schlüssel nichts anfangen kann. Auch lässt sich nicht feststellen, ob der Nutzer auch einen anderen Dienst benutzt, da für jeden Dienst ein neues Schlüsselpaar erstellt wird.

Der Nachteil ist wie bei SQRL, dass der Nutzer sich selbst um die Sicherheit seines FIDO Authenticators kümmern muss. Auch kann der Dienst nur den Authenticator authentifizieren und nicht den Nutzer identifizieren.

4.14 PseudoID

PseudoID [10] ist ein Protokoll mit einem ganz besonderen Zweck: Schutz der Privatsphäre von Nutzern (englisch: Privacy Enhancing IDM). Der Identitätsprovider soll hier nicht darüber informiert werden, auf welchen Seiten sich der Nutzer anmeldet. Der Identitätsprovider besteht daher aus zwei Komponenten. Der erste Teil ist der eigentliche Identitätsprovider, der Nutzeridentitäten gegenüber Diensten bestätigt. Die zweite Komponente ist der Blind Signature Service (BSS). Dieser Service übernimmt den Part der Authentifikation. IdP und BSS müssen nicht zur selben Domäne gehören. Wenn der Nutzer sich bei einem Dienst anmelden möchte, wird er wie gewohnt zu seinem Identitätsprovider weitergeleitet. Von dort folgt eine Weiterleitung zum BSS. Der nachfolgende Ablauf ist in Abbildung 4.14 dargestellt. Beim BSS meldet sich der Nutzer an und sendet gleichzeitig ein Blinded Token mit (1). Dieses Token enthält den gewünschten Nutzernamen bzw. Pseudonym für diesen Dienst uns ist kryptografisch verschlüsselt, sodass der BSS diesen Namen nicht lesen kann. Nach erfolgreicher Anmeldung signiert der BSS das Token und schickt es an den Nutzer zurück (2). Im dritten Schritt (3) wird das Token wieder entschlüsselt. Das hier eingesetzte kryptografische Verfahren basiert auf Blinden Signaturen [6]. Die Signatur bleibt auch für den unverschlüsselten Namen weiterhin gültig. Das so signierte Pseudonym ist die PseudoID. Die PseudoID oder auch das Access Token (AT) kann nun zum Identitätsprovider geschickt werden (4). Beim IdP wird die Signatur geprüft (5). Danach kann die Identität dem Dienst bestätigt werden und der Nutzer erhält Zugriff.

Vorteil von PseudoID ist, dass weder BSS noch IdP wissen, welche Dienste von welchem Nutzer benutzt werden. Der BSS kennt den Nutzer, aber nicht das Pseudonym und der IdP kennt das Pseudonym, aber nicht den Nutzer.

Als Nachteil ist die fehlende Semantik der PseudoID zu nennen. Die ID enthält kaum weitere Attribute. Der IdP kann so nicht entscheiden, ob er diese ID bereits

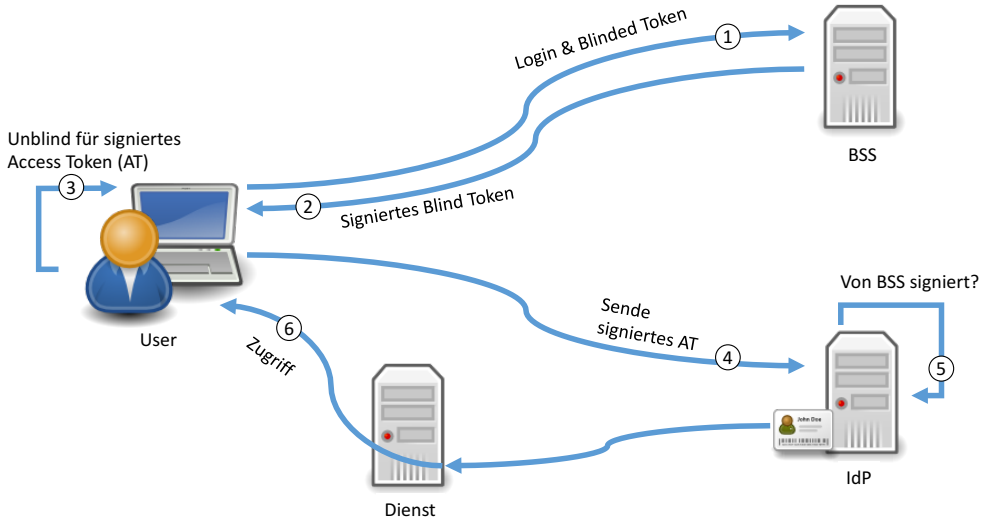

Abbildung 4.14: PseudoID – Workflow

für einen Dienst bestätigt hat oder nicht. Eine abgefangene PseudoID kann vom Angreifer wieder verwendet werden (Replay Angriff[6]).

4.15 BlindIDM

BlindIDM [21] ist ein Modell, das wie PseudoID die Privatsphäre des Nutzers schützt. Auch hier weiß der IdP nicht, welcher Nutzer sich bei welchem Dienst anmeldet. Die Ausgangssituation ist die, dass eine Organisation (Host) das Identitätsmanagement an einen Identitätsprovider in der Cloud auslagert. Die Nutzerdaten bleiben aber beim Host. Das bedeutet, dass der Host die Authentifikation übernimmt und der IdP die Identitätsbestätigung bei den Diensten. Der Ablauf ist in Abbildung 4.15 dargestellt. Wie immer, wenn der Nutzer einen Dienst benutzen will (1), wird er zu seinem Identitätsprovider weitergeleitet (2) und von dort dann zum Host (3). Dort meldet er sich mit seinen Daten an (4). Der Host verschlüsselt das Identitätsdokument mit seinem öffentlichen Schlüssel. Das verschüsselte Token wird zum IdP hochgeladen (6). Dort wird das Token umgewandelt durch Proxy Re-Encryption (PRE) (7). Dann wird das Token zum Dienst übermittelt (8), der es mit dem privaten Schlüssel entschlüsseln kann (9). Jetzt kann dem Nutzer Zugang zum Dienst gegeben werden. Proxy Re-Encryption ist eine kryptografische Methode, die einen kodierten Text unter einem Schlüssel A zu einem kodierten Text unter Schlüssel B umwandelt, ohne die beiden Schlüssel zu kennen. In diesem Fall

[6] `https://cwe.mitre.org/data/definitions/294.html`, besucht am 16.02.2017.

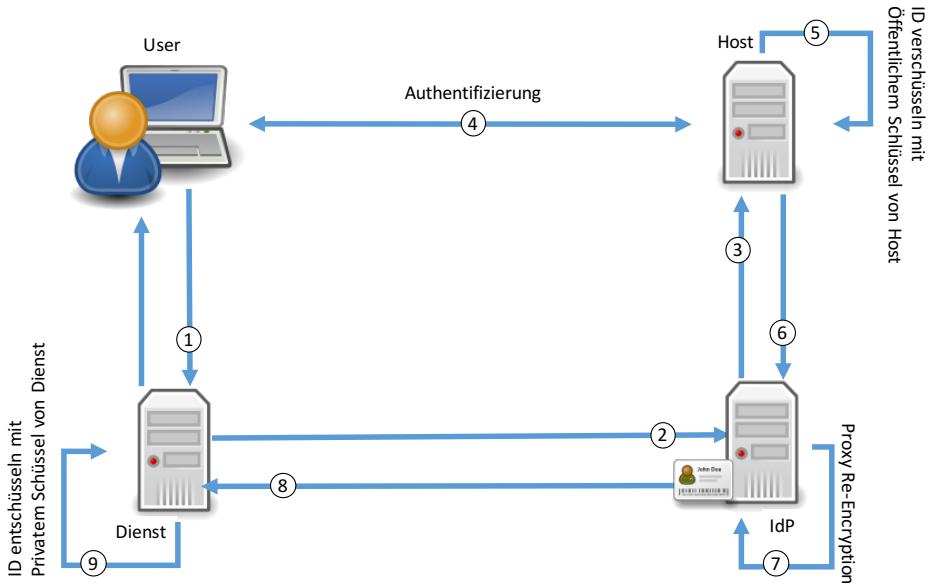

Abbildung 4.15: BlindIDM – Workflow mit Proxy Re-Encryption

wird die Umwandlung vom IdP übernommen. Damit er das kann, braucht er eine entsprechende Funktion. Die wird vom Host berechnet, da dieser die notwendigen Informationen hat: den privaten Schlüssel vom Host zum Entschlüsseln und den öffentlichen Schlüssel des Dienstes zum Verschlüsseln. Den öffentlichen Schlüssel bekommt der Host, wenn er den Dienst als vertrauenswürdig ansieht.

Vorteil von BlindIDM ist zum Einen, dass Host und IdP nicht wissen, welcher Nutzer welchen Dienst benutzt. Zum Anderen kann der IdP von Dritten betrieben werden, da er nicht in der Lage ist das ID Token zu entschlüsseln.

Ein Nachteil ist, dass ein BlindIDM System nicht sofort mit jedem Dienst funktioniert. Der Host muss den Dienst erst als Vertrauenswürdig einstufen um die entsprechende Funktion für den IDP zu berechnen. Erst dann kann ein Nutzer sich über seinen Host bei dem Dienst anmelden.

4.16 Diskussion

Die im Abschnitt 1.3 vorgestellten Ansätze des Managements digitaler Identitäten verwenden verschiedene Protokolle. Tabelle 4.1 zeigt, welches der Protokolle zu welchem Ansatz gehört. Danach wird noch auf Herausforderungen eingegangen, die trotz vorhandener Produktlösungen weiterhin existieren. Schließlich soll noch die Notwendigkeit der Kombination der einzelnen Protokolle behandelt werden.

Protokoll	IDM Ansatz
Kerberos	zentral
PKI	zentral
WS-* Spezifikationen	föderiert
WebID	dezentral
Persona (BrowserID)	dezentral
OpenID	dezentral
OAuth	zentral
OpenID	zentral
UMA	zentral
SAML	zentral
SCIM	dezentral
SQRL	dezentral
FIDO	dezentral
PseudoID	zentral
BlindIDM	zentral

Tabelle 4.1: Zuordnung der IdM Protokolle zu den Identitätsmanagement-Ansätzen

4.16.1 Herausforderungen von Identitätsmanagement und Single-Sign-On

Obwohl es bereits viele Technologien und Protokolle für das Identitätsmanagement gibt, werden sie noch längst nicht überall angewendet. Andererseits erlauben immer mehr Dienste ein einfaches Login per sozialem Netzwerk wie Facebook oder Google. Sobald aber ein Dienst sensible Daten benötigt, z.B. die Adresse für den Versand von Waren oder Kontoinformationen für den Bezahlvorgang, sind soziale Netzwerke nicht mehr ausreichend. Der Grund liegt in der Frage der Glaubwürdigkeit, denn solche Daten können etwa bei Facebook gar nicht oder falsch eingegeben werden. Deshalb müssen Nutzer bei dem entsprechenden Dienst doch einen neuen Account anlegen und die benötigten Daten separat eingeben. Ein hoheitlicher Identitätsprovider wäre für solch eine Aufgabe besser geeignet, da ihm die Anschriften bekannt wären und er auch einen entsprechenden Vertrauensgrad hätte.

Ein Nutzer muss auch dann einen weiteren Account anlegen, wenn ein Dienst mit Single-Sign-On nur einen bestimmten Identitätsprovider akzeptiert, bei dem er aber keinen Account hat – etwa wenn ein Dienst nur Logins mit Facebook akzeptiert, der Nutzer jedoch nur ein Google-Konto hat.

Die Verwendung eines einzigen Identitätsproviders für alle Dienste erfordert einen besonders hohen Schutz des IdP-Kontos. Die meisten Identitätsmanagement-Systeme im Internet implementieren jedoch nur die Username/Passwort-Methode. Doch diese kann, wie erläutert, Sicherheitsrisiken mit sich bringen.

Es gibt deshalb immer mehr Dienste, die ihre Authentifizierungsmethode erweitern, um die Konten der Nutzer besser zu schützen. Ein Beispiel ist Google: Nutzer haben die Möglichkeit, ein USB-Token zum Schutz des Kontos zu verwenden oder sie geben einen Code ein, der vom Smartphone und der passenden App generiert wird. Durch den zweiten Faktor steigt allerdings der Aufwand des Nutzers, den er zur Authentifikation zu leisten hat.

Sun et al. [25] fanden heraus, dass viele Nutzer das Single-Sign-On noch nicht richtig verstehen. Danach haben manche Angst davor, dass ihre Zugangsdaten vom IdP an den Dienst weitergeben werden oder sorgen sich, die Kontrolle über ihre Daten und Accounts zu verlieren. Auch fürchten sie den potenziellen „Single Point of Failure"(beim Ausfall des IdP kann man sich bei keinem Dienst mehr anmelden). Zudem wissen viele Nutzer nicht, welche Informationen wirklich zwischen IdP und SP ausgetauscht werden [1]. Durch Protokolle wie OAuth jedoch werden die Nutzer über die Daten, die angefragt werden, informiert und sie müssen aktiv zustimmen. Oftmals gibt es nur die Optionen des Zustimmens oder des Ablehnens. Das An- oder Abwählen einzelner Attribute ist selten möglich.

Ein potenzielles Problem bei den meisten IdM-Systemen ist auch, dass der IdP immer weiß, bei welchen Diensten sich welcher Nutzer angemeldet hat. Das geschieht durch das Umleiten (redirecting) zwischen Dienst und IdP. Es gibt unter den vorgestellten Ansätzen jedoch einige, die das verhindern und so die Privatsphäre etwas stärker schützen. Dazu gehören BrowserID, PseudoID und BlindIDM, die jedoch in der Praxis kaum Anwendung finden.

Es kann aber auch ganz auf den IdP verzichtet werden wie beim Einsatz von SQRL oder FIDO.

Auf der anderen Seite ist es für einige Nutzer vorteilhaft zu wissen, wo man sich bereits angemeldet hat. Dadurch kann man Diensten, denen man Zugriff auf die Identitätsdaten gegeben hat, diesen auch wieder entziehen.

4.16.2 Kombination von Technologien

Um ein modernes Identitätsmanagement zu schaffen, reicht es nicht mehr aus, nur einen einzigen Standard zu wählen und diesen zu implementieren. Denn für verschiedene Aufgaben werden unterschiedlich geeignete Protokolle benötigt. Es braucht auf jeden Fall eine solide Basis, um eine Identitätsmanagementplattform für das Internet zu bauen, die ein breites Spektrum an Aufgaben erfüllen kann. Für jede Aufgabe eignet sich jeweils mindestens ein Protokoll besonders. Ein Beispiel für eine solche Protokollsuite zeigt der Neo Security Stack[7] in Abbildung 4.16.

Für die Authentifikation von Nutzern bietet sich FIDO besonders an. FIDO erlaubt ganz unterschiedliche Methoden, die sich auch kombinieren lassen, um die Sicherheit zu erhöhen. Neue Methoden können einfach hinzugefügt werden, ohne das Kommunikationsprotokoll ändern zu müssen. Um Nutzeridentitäten oder

[7] `http://nordicapis.com/api-security-oauth-openid-connect-depth`, besucht am 16.02.2017.

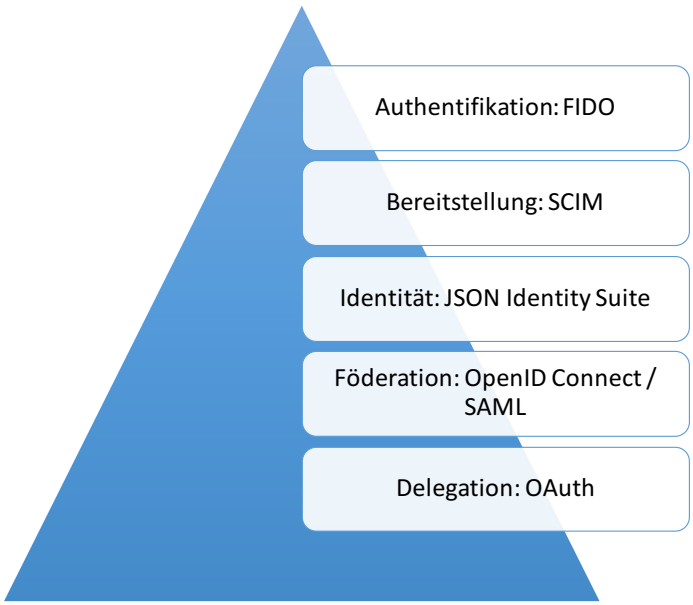

Abbildung 4.16: NEO-Security Stack

Gruppen zwischen Diensten anzulegen, zu lesen oder zu löschen, kommt der SCIM-Standard in Frage. Damit lassen sich auch Identitäten bei anderen Diensten automatisch verwalten. Als Repräsentation von Identitäten und anderen Daten ist JSON eine gute Wahl. Es ist ein einfach lesbares und kompaktes Format, es wird auch von SCIM, OIDC und auch vielen anderen Diensten verwendet und mit entsprechenden Frameworks (z.B. JOSE) können die Daten auch verschlüsselt und signiert werden. Damit Dienste Identitäten aus anderen Domänen akzeptieren können, muss der Dienst in der Lage sein, die erhaltene Identität zu überprüfen. Dafür eignen sich die Protokoll OpenID Connect und SAML: Hier werden signierte Identitätstoken spezifiziert, um eine Förderation bzw. ein offenes Identitätsmanagement zu erreichen.

Neben Identitätsdaten werden auch andere Daten von Nutzern verwaltet. Nutzer teilen gerne ihre Daten mit Freunden und Verwandten. Es ist möglich, diesen Menschen das Recht zu geben, auf bestimmte Daten zugreifen zu dürfen. Hier bedarf es eines Autorisationsprotokolls – die ideale Anwendung für OAuth.

All diese Protokolle bilden eine gute Suite, die Grundlage für sichere IdM-Systeme sein kann.

5 Forschung am Hasso-Plattner-Institut

In diesem Abschnitt werden abgeschlossene und laufende Forschungsarbeiten des Hasso-Plattner-Instituts zum Thema Identitätsmanagement vorgestellt.

5.1 Bisherige Forschungsarbeiten

Bereits in der Vergangenheit hat das HPI Forschung zu Identitätsmanagement-Themen betrieben. Einige dieser Forschungsarbeiten und ihre Ergebnisse werden in diesem Abschnitt vorgestellt.

5.1.1 Glaubwürdigkeit auf der Ebene von Attributen

Im zentralisierten, dezentralisierten und föderierten Ansatz wird die Aufgabe der Überprüfung und Bereitstellung einer Identität auf einen Identitätsprovider übertragen. Der Provider muss nicht zur selben Domäne wie der Dienst gehören. Damit ein Dienst die Identität, die der Identitätsprovider erstellt hat, akzeptiert, muss er dem Provider vertrauen. Daraus folgt, dass alle Attribute, welche die Identität bilden, als glaubwürdig angesehen werden. Wenn Attribute beim Identitätsprovider nicht verifiziert werden, kann ein Nutzer auch falsche Angaben bezüglich seiner Identität machen. Diese falschen Informationen werden dann an Dienste weitergegeben, wo sie als glaubwürdig anerkannt werden. Deshalb sollen Dienste die benötigten Informationen von unterschiedlichen Providern bekommen. Abbildung 5.1 soll das an einem Beispiel verdeutlichen [27]. Eine Zeitung bietet ein günstiges Abonnement für Studierende an. Diese müssen dafür ihre Anschrift, Kontonummer und einen Studierendenstatus angeben. Bei der Universität, dem Bürgeramt und der Bank hat jeder studentische Nutzer bereits Konten, die Teilinformationen zu seiner Identität enthalten. Statt alle Informationen bei der Zeitung einzugeben, kann der Nutzer auf andere Konten verweisen, die bereits diese Informationen haben: Die Kontoinformation kann direkt bei der Bank eingeholt werden und der Studentenstatus bei der Universität. Damit die Zeitung bzw. ein Dienst entscheiden, ob die jeweiligen referenzierten Attribute glaubwürdig sind, haben Thomas et al. [27] ein Modell entworfen, mit dem diese Glaubwürdigkeit von Attributen beschrieben werden kann. Zusätzlich zu jedem Attribut wird gespeichert, ob es verifiziert ist oder nicht. Die Art der Verifizierung hängt dabei von den Attributen ab. Zum Beispiel können Name und Adresse über ein offizielles Dokument wie den Personalausweis verifiziert werden. Für den Dienst ergibt sich dann die Glaubwürdigkeit eines Attributes daraus, dass der IdP vertrauenswürdig und das Attribut

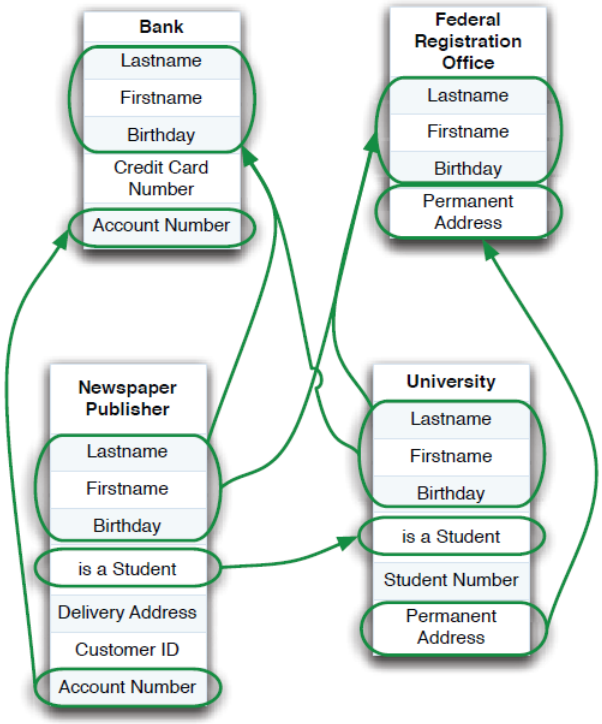

Abbildung 5.1: Beispiel zum Vertrauen auf dem Level von Attributen

verifiziert ist. Insgesamt soll so auch die Redundanz von Identitätsdaten verringert werden, in dem einfach auf bereits einmal eingegebene Daten referenziert wird.

5.1.2 HPI Identity Provider

Im Rahmen der Forschung mit Attributen und deren Verifizierung ist zudem der HPI Identity Provider[1] entstanden. Er basiert auf den offenen Web-Standards WS-Trust, SAML, Information Card und OpenID. WS-Trust spezifiziert den Security Token Service (STS). Dieser Service dient dazu, Identitätstokens auszustellen und digital zu signieren. Das Token basiert auf SAML. Die generelle Identitätsmanagement-Funktionalität wird mit OpenID realisiert. Nutzer können Accounts erstellen, bearbeiten und löschen. In einem Account können mehrere Identitäten verwaltet werden. Zu jedem Attribut wird gespeichert, ob es verifiziert wurde oder nicht. Zum Beispiel kann die Verifikation der E-Mail-Adresse mit einer Bestätigungs-E-Mail erreicht werden: Der Nutzer klickt dann auf einen Link innerhalb dieser E-Mail. Dieses Verfahren wird bei vielen Diensten eingesetzt, um die E-Mail-Adresse zu bestätigen. Es ist auch möglich, die E-Mail-Adresse mit einem gültigen und akzep-

[1] `https://openid.hpi.uni-potsdam.de`, besucht am 16.02.2017.

tierten Zertifikat zu bestätigen. Der Verifikationsstatus wird dann zusammen mit dem Wert eines Attributes vom Identitätsprovider zu einem Dienst übertragen.

Das SAML-Token kann nicht ohne Weiteres neue Attribute wie den Verifikationsstatus aufnehmen. Deshalb haben Thomas et al. [29] das Schema von SAML erweitert, um neue Metadaten wie den Verifikationsstatus oder auch die Verifikationsmethode aufnehmen zu können. Der HPI Identity Provider ist nicht nur für die Forschung zugänglich. Mitarbeiter und Studenten des Instituts können ihn jederzeit verwenden, um zum Beispiel OpenID unterstützende Services zu nutzen.

5.1.3 Vergleich von Authentifikationsmethoden

Es gibt bereits zahlreiche Authentifikationsmethoden. Deshalb fällt es oft schwer zu ermitteln, welche Methode sicherer ist. Ein Beispiel: Ein Dienst, etwa der einer Online-Bank, verlangt eine sichere Anmeldung und verwendet die Zwei-Faktor-Authentifizierung mit Passwort und Smartphone-Code. Mit FIDO UAF ist es möglich, dem Nutzer unterschiedliche Methoden anzubieten, je nach dem, was der Authentifikator unterstützt. Die Bank möchte ihren Kunden die Wahl der Methode selbst überlassen, solange diese Methode mindestens so sicher wie die bisherige Zwei-Faktor-Authentifikation ist. Wie solche Methoden vergleichbar gemacht werden können, zeigt die Arbeit von Thomas et al. [30]. Der Vergleich zweier Methoden erfolgt über die Berechnung eines Trustlevels (Vertrauenslevel). Dieses Level bestimmt sich daraus, wie wahrscheinlich es ist, das die Methode von einem Angreifer A „geknackt“ wird. Diese Wahrscheinlichkeit kann entweder berechnet werden oder muss empirisch ermittelt werden. Die Formel aus der Arbeit lautet:

$$-\log(\Pr(A))$$

Für eine kombinierte Methode kann das Trustlevel wie folgt bestimmt werden:

$$-\log(\Pr(A \cap B))$$

Dabei ist A die Wahrscheinlichkeit, dass die erste Methode ausgehebelt wird und B die Wahrscheinlichkeit, dass die zweite Methode gebrochen wird.

Mit dieser Metrik kann die Bank festlegen, dass alle Authentifizierungsmethoden erlaubt sind, die dasselbe oder ein höheres Trustlevel haben als die bisher benutzte 2FA.

5.2 Aktuelle Forschungsprojekte

5.2.1 Identity Leak Checker

Identitätsdiebstähle haben in den vergangenen Jahren zugenommen. Deshalb hat das Hasso-Plattner-Institut mit dem HPI Identity Leak Checker[2] einen Internet-

[2] `https://sec.hpi.de/leak-checker`, besucht am 16.02.2017.

dienst geschaffen, der Personen die Möglichkeit bietet, zu überprüfen, ob ihre persönlichen Daten bei größeren Leaks betroffen worden sind. Der Identity Leak Checker durchsucht das Internet nach unrechtmäßig veröffentlichten Identitätsdaten und stellt die frei zugänglichen Daten in einer Datenbank zusammen [18]. Zu den betroffenen Informationen gehören unter anderem Passwörter, Vor- und Zunamen, Anschriften, Konto- und Kreditkartendaten und Geburtstage. Derzeit sind bereits mehr als 3 Milliarden Identitäten erfasst worden.

Über eine Web-Oberfläche können mögliche Opfer eines Identitätsdiebstahls überprüfen, ob ihre Daten in der Datenbank auftauchen. Hierfür geben sie ihre E-Mail-Adresse in ein Suchfeld auf der Webseite ein und erhalten im Anschluss eine Benachrichtigung, ob und wenn ja welche veröffentlichten Informationen im Zusammenhang mit der E-Mail-Adresse gefunden wurden. Die Antwortmail enthält im Fall der Betroffenheit Handlungsempfehlungen, die den Opfern helfen sollen, Schutzmaßnahmen gegen Diebstahl zu ergreifen, die einen weiteren Missbrauch verhindern oder zumindest erschweren.

Der Identity Leak Checker erstellt außerdem umfangreiche Statistiken auf Grundlage der in der Datenbank gespeicherten und anonymisierten Identitätsdaten. Diese Statistiken umfassen neben einer Übersicht, wie viele Identitäten im Laufe der vergangenen Jahre veröffentlicht wurden, auch Statistiken hinsichtlich der meistgenutzten Passwörter und die Verteilung der Domänen für alle erfassten E-Mail-Adressen.

5.2.2 Verhaltensbasiertes Authentifizieren mit Smartphone und Wearables

Wie in Sektion 3 gezeigt, gibt es viele Verfahren der Authentifikation, die alle bestimmte Schwächen aufweisen. Ein-Faktor-Authentifikation wird selten richtig angewandt. Die Kombination von Faktoren zur Mehr-Faktor-Authentifikation jedoch verstärkt die Sicherheit. Nutzer müssen allerdings mehr Schritte durchführen, um sich erfolgreich zu authentisieren. Das wirkt sich negativ auf die Bedienbarkeit aus. Nutzer wollen komfortable Dienste nutzen und sich nicht ausführlich mit der Authentifikation befassen.

In einer Forschungsarbeit wird die in Abschnitt 3.4 beschriebene Idee aufgenommen und weiterverfolgt, dass Computer ihre Besitzer erkennen. Die Computer sollen Menschen genauso erkennen, wie es andere Menschen tun: anhand ihrer körperlichen Merkmale und ihres Verhaltens. Wie im Abschnitt 3.3 zur Biometrie erwähnt, bedarf es bei solchen Verfahren meistens spezieller teurer Lesegeräte bzw. Hardware. Das macht sie für Nutzer weniger attraktiv. Deshalb sollten besser Geräte zum Einsatz kommen, die die Nutzer bereits haben. Das ist, allem voran, das Smartphone. Über 80 Prozent der Internetnutzer verwenden ein Smartphone.[3] Auch Wearables wie Fitness-Tracker und Smart Watches werden immer beliebter.

[3] `http://www.globalwebindex.net/blog/80-of-internet-users-own-a-smartphone`, besucht am 16.02.2017.

Diese Geräte sind mit vielen Sensoren, die Daten erfassen können, ausgestattet. Wearables haben den Vorteil, dass sie direkt am Körper getragen werden und so bessere Daten liefern können. Mithilfe dieser Daten lassen sich biometrische Merkmale berechnen und dem Besitzer zuordnen.

Jedes Gerät berechnet seine eigenen Merkmale und versucht selbst, seinen eigenen Besitzer zu erkennen. Das Ergebnis wird als Vertrauens-Niveau oder auch Trust Level dargestellt. Dieses Level gibt die Wahrscheinlichkeit an, mit der das Gerät sicher geht, dass der aktuelle Nutzer auch der Besitzer ist. Das Vertrauens-Niveau wird an das Smartphone gesendet. Hier werden alle Level aller eingesetzten Wearables gesammelt. Zusätzlich berechnet das Smartphone sein eigenes Level. Das Gesamtergebnis kann nach außen weitergegeben werden. Es wird nur der Wert des Vertrauens-Niveaus weitergegeben; die biometrischen Daten bleiben aus Sicherheitsgründen selbstverständlich auf den Geräten des Besitzers und verlassen diese nicht.

Das Vertrauens-Niveau kann dann an einen Identitätsprovider geschickt werden. Dieser wiederum kann das Trust Level dann an alle vom Nutzer verwendeten Dienste weiterleiten. Diese können dann entscheiden, welche Rechte sie dem Benutzer geben - basierend auf dem Vertrauens-Niveau. So kann jeder Dienst selbst entscheiden, bei welchem Level er den Nutzer als authentifiziert ansieht. Für ein Diskussionsforum kann das bei 75 Prozent der Fall sein, während eine Bank mindestens 90 Prozent haben will. Durch kontinuierliche Berechnung der Trust Level kann schnell festgestellt werden, wenn sich der Nutzer und sein Verhalten ändert. Das Vertrauens-Niveau sinkt dann; es wird an den Identitätsprovider geschickt, der dann alle verwendeten Dienste informieren kann oder sofort bei sich und allen Diensten ein Logout einleitet.

Dieser Ansatz bietet gute Sicherheit durch die Kombination biometrischer Merkmale mit dem Besitz von Smartphones und Wearables. Um das Nutzerverhalten zu bestimmen, werden kontinuierlich Daten erfasst und ausgewertet. Das geschieht im Hintergrund. Wenn der Nutzer sich anmelden will, brauchen diese Daten nur noch abgefragt werden. Das ist bequem für den Nutzer. Er muss dann zum Beispiel auf seinem Smartphone nur noch ein OK drücken. Das ist schon alles. Er braucht sich nichts zu merken, sondern kann sich wie immer verhalten. Die Authentifikation wird dadurch für ihn leicht und für andere schwer, da diese ein anderes Verhalten haben. Änderungen im Nutzungsverhalten werden schnell registriert. Unbefugten kann direkt der Zugang zum Smartphone und zu den Diensten gesperrt werden.

Zu den ersten Faktoren, die bei einer prototypischen Anwendung erfasst wurden, gehören das Gangverhalten, verbundende WLAN-Netzwerke, bekannte geographische Orte und die typische Bewegung, mit der das Smartphone aus der Tasche gezogen wird. Die Gangerkennung kann von Smartphones und Wearables durchgeführt werden. Der Gang von Menschen unterscheidet sich durch Eigenschaften wie Schrittgeschwindigkeit, Schrittlänge oder Hüftschwung signifikant. Diese Parameter können mit dem Beschleunigungssensor und Gyroscope erfasst werden. Mit diesen Sensoren kann aber noch viel mehr erfasst werden, zum Beispiel auch die Bewegung, mit der das Smartphone hervorgeholt wird. Diese Bewegung kann noch besser bestimmt werden, wenn auch ein Lichtsensor zum Einsatz kommt.

Ferner kann durch WLAN und GPS festgestellt werden, ob sich das Gerät in einem bekannten Umfeld wie der Wohnung oder dem Arbeitsplatz befindet. Authentifizieren von einem unbekannten Ort aus, etwa nach einem Diebstahl, wird das Vertrauens-Niveau verringern.

Geplant ist, weitere verhaltensbasierte Faktoren hinzuzufügen. Es sollen auch ganze Routinen erfasst und verglichen werden. Die meisten Menschen folgen bestimmten täglichen Routinen: Sie stehen zur selben Zeit auf, fahren mit Bus, Bahn oder Auto zur Arbeit usw. Durch Analyse dieser Aktivitäten lassen sich Tages-, Woche- oder Wochenend-Routinen ermitteln. Diese können dann mit den Livedaten verglichen werden um zu prüfen, ob sie in die Muster passen.

6 Fazit

Die vorliegende Studie will deutlich machen, wie wichtig Identitätsmanagement ist. Bisherige Lösungen, vor allem solche mit Passwort-Authentifikation, weisen oftmals Probleme auf. Es werden viele Angriffe auf Dienste und deren Nutzer ausgeführt, um sensible Daten wie Passwörter zu stehlen. Der Nutzer selbst bekommt eine Attacke, wenn überhaupt, oft erst sehr spät mit und kann dann nur noch reagieren. Wie gezeigt, gibt es für das Authentifizieren mit Passwörtern Alternativen und Optionen, die jeweils bestimmte Vor- und Nachteile haben. Ferner existieren Technologien und Standards, deren Anwendung dazu führt, dass weniger Konten nötig sind und somit auch weniger Passwörter. Jedoch werden solche Systeme nur selten eingesetzt. Außnahmen bilden die sozialen Netzwerke wie Facebook, die eine Anmeldung bei vielen Diensten ermöglichen.

Aktuelle Identitätsmanagementsysteme funktionieren gut, so lange Nutzer sich innerhalb einer Domäne bewegen: Ein Gmail-Konto bei Google erlaubt z. B. gleichzeitig auch, ohne zweite Anmeldung den YouTube-Dienst zu verwenden. Ähnlich ist es bei Apple; wenn alle Geräte von diesem Hersteller stammen, passt alles. Soll aber das Gerät eines fremden Herstellers mit einem von Apple kommunizieren oder Daten austauschen, bedarf es oftmals komplizierter Abläufe.

Desweiteren haben Identitätsmanagementsysteme Probleme mit Datenschutz und Selbstbestimmung: Daten werden unerlaubt erhoben, teils mehr als nötig, teils im Hintergrund, also ohne dass die meisten Nutzer etwas davon mitbekommen. Zusätzlich werden Daten unerlaubt weitergeben. Für bestimmte Funktionen müssen Daten auf Ressourcen irgendwo im Internet, also in der Cloud, verarbeitet werden, da lokale Kapazitäten nicht ausreichend sind. Auch hier müssen Daten preisgegeben werden. Nutzer haben keine Kontrolle über ihre eigenen Daten bzw. über die eigene Identität.

Die Zukunft bringt jedoch neue Herausforderungen. Die Komplexität des Datenaustauschs wird zunehmen. Nutzer üben verschiedene Rollen gleichzeitig aus, sind sowohl privat aktiv, als auch als Bürger im Kontakt mit Behörden und in der Wirtschaft als Beschäftigte und Verbraucher. Die unterschiedlichen Rollen führen jeweils zu unterschiedlichen Attributen, lassen sich aber nicht strikt trennen. Ein Beispiel: Reisen in der Rolle des Beschäftigten eines Wirtschaftsunternehmens werden durchaus zunächst mit der privaten Kreditkarte bezahlt. Die Kosten werden dann später durch das Unternehmen erstattet.

Zukünftige Identitätsmanagementsysteme müssen mit den verschiedenen Rollen umgehen können. Dabei sollen sie stets sicher und komfortabel in der Nutzung sein. Viele Menschen haben immer engere Zeit-Budgets und wollen nicht mit umständlicher Authentifikation Zeit verschwenden. Sie erwarten Bequemlichkeit und

Flexibilität. Dazu müssen sie allerdings auch Daten selbst freigeben – selbstbestimmt und mit voller Kontrolle.

Einen ersten Schritt in diese Richtung macht die verhaltensgestützte Analyse und Authentifikation. Sie ist sehr bequem, da der Nutzer sich kein Geheimnis einprägen oder etwas mitbringen muss – mit Ausnahme des Smartphones, das aber für über 60 Prozent der Nutzer nicht mehr wegzudenken ist.[1] Die meisten Daten werden im Hintergrund verarbeitet. Dadurch, dass jeder Mensch ein anderes, typisches Verhalten zeigt, bietet dieses Verfahren sehr guten Schutz. Durch immer bessere Sensorik, die feiner zu messen hilft, wird eine Nachahmung individuellen Verhaltens künftig weiter erschwert werden. Der verhaltensbasierte Ansatz findet nicht nur online Anwendung für Web-Applikationen, sondern kann auch offline zum Tragen kommen, zum Beispiel für das Öffnen von Türen und Zugängen. Dank Sensoren, die ihre Werte in das Internet der Dinge funken, kann die eigene Tür den Bewohner bzw. Mitarbeiter erkennen und ihn einlassen.

Mit Hilfe des Verhaltens lassen sich nicht nur Menschen erkennen, sondern es können auch Dinge wie Maschinen und Materialien eindeutig identifiziert werden. Schließlich verhalten sich z. B. Stoffe bei bestimmten Temperaturen spezifisch. Das können Sensoren ebenfalls messen und übermitteln.

So bieten die verhaltensgestützten Methoden des Managements digitaler Identitäten viele Möglichkeiten, um die Herausforderungen der Zukunft anzupacken.

[1] `https://www.bitkom.org/Presse/Presseinformation/Nicht-ohne-mein-Smartphone.html`, besucht am 16.02.2017.

Glossar

ADFS Active Directory Federation Services, ein von Microsoft entwickeltes WS-* Single-Sign-On System.

API Application Programming Interface, eine Schnittstelle für Programmierer.

Assertion Eine Behauptung oder Claim im SAML-Protokoll.

Asymmetrische Verschlüsselung Ein Verschlüsselungssystem mit zwei Schlüsseln, einen öffentlichen Schlüssel zum Verschlüsseln und einen privaten Schlüssel zum Entschlüsseln.

Attribut Ein Merkmal oder Eigenschaft einer Entität.

Authentifizierung Eine Prüfung eines Identitätsnachweises, wird vom Dienst ausgeführt und folgt meistens nach einer Authentisierung.

Authentisierung Ein Nachweis der Identität, wird vom Benutzer ausgeführt. Z.B. durch Vorlegen eines Ausweisdokumentes oder Eingabe eines Passwortes.

Autorisierung Eine Prüfung der Rechte einer Entität in einem System.

Biometrie Eine Bezeichnung für Methoden, die Entitäten anhand ihrer physischen Merkmale oder ihrem Verhalten identifizieren können.

BitID Ein Authentifikationsprotokoll in Bitcoin-Systemen, dass ähnlich zu SQRL ist und QR-Codes verwendet.

BlindIDM Ein Identitätsmanagement-Protokoll bei dem der Identitätsprovider an einen dritten Dienst ausgelagert werden kann.

BrowserID Siehe Mozilla Persona.

BSS Blind Signature Service, eine Komponente im PseudoID-Protokoll.

CA Certificate Authority, eine Zertifizierungsstelle in einer PKI.

Claim Eine Behauptung über eine Entität.

Entität Ein Objekt in der Welt, z.B. Person oder Computer.

FIDO Fast IDentity Online, eine Bezeichnung für die Authentifizierungsprotokolle U2F und UAF der FIDO Alliance.

FOAF Friend Of A Friend, ein Vokabular für RDF.

HTTP HyperText Transfer Protocol, ein Kommunikationsprotokoll im Internet.

HTTPS HyperText Transfer Protocol Secure, eine sichere Variante von HTTP.

Identität Eine Zusammenfassung aller Merkmale zur Charakterisierung einer Entität.

Identitätsdiebstahl Eine Entität benutzt unerlaubt eine nicht zu ihm gehörende Identität.

Identitätsmanagement Ein Identitätsmanagementsystem beschäftigt sich mit der Verwaltung, Authentifizierung und Autorisierung von Identitäten.

Identitätsprovider Ein spezieller Dienst, der Identitätsinformationen von Nutzern speichert und diese gegenüber anderen Diensten bereitstellt und bescheinigt.

IDM Siehe Identitätsmanagement.

IdP Siehe Identitätsprovider.

JOSE Javascript Object Signing and Encryption, ein Framework zum Verschlüsseln und Signieren von JWTs.

JSON JavaScript Object Notation, ein einfaches, kompaktes und lesbares Datenformat.

JWT JSON Web Token, ein auf JSON basierender Standard zum Erstellen von Tokens.

KDC Key Distribution Center, ein Identitätsprovider im Kerberos-Protokoll.

Keylogger Ein Programm zum Aufzeichnen von Tastatureingaben.

LDAP Lightweight Directory Access Protocol, ein Protokoll für verteilte Verzeichnisse zum Abrufen und Ändern von Daten in einem Netzwerk.

Leak Eine unerlaubte Veröffentlichung von Daten.

Mozilla Persona Ein Authentifikationsprotokoll, bei dem die E-Mail-Adresse als Identität und Zertifikate als Token verwendet werden.

OAuth Ein Autorisierungsprotokoll, dass es einem Nutzer erlaubt, anderen Nutzern Zugang zu eigenen Daten zu geben, ohne die eigenen Login-Informationen preiszugeben.

OIDC Siehe OpenID Connect.

OpenID Ein Authentifikationsprotokoll, bei dem eine URL die Identität repräsentiert.

OpenID Connect Ein Authentifikationsprotokoll und Erweiterung von OAuth. Es gibt zusätzlich ein Identitätstoken.

OpenPGP Open Pretty Good Privacy, ein offener Standard zum Verschlüsseln von E-Mails.

OTP One Time Password, ein Passwort, dass nur einmal verwendet wird und danach ungültig ist.

Password Phishing Ein Angriff, bei dem durch gefälschte E-Mails, Webseiten oder Textnachrichten der Benutzer dazu verleitet wird, sein Passwort preiszugeben.

Password Sniffing Ein Angriff, bei dem Datenpakete im Netzwerk nach Passwort-Informationen durchsucht werden.

PKI Siehe Public-Key-Infrastruktur.

PRE Proxy Re-Encryption, eine Transformation von verschlüsselten Texten unter einem Schlüssel A zu einem anderen Schlüssel B ohne vorher zu entschlüsseln.

Private Key Siehe Privater Schlüssel.

Privater Schlüssel Ein Teil eines asymmetrischen Verschlüsselungsverfahrens. Damit wird eine Nachricht signiert oder entschlüsselt.

PseudoID Ein Authentifikationsprotokoll, das mit Pseudonymen arbeitet.

Public Key Siehe Öffentlicher Schlüssel.

Public-Key-Infrastruktur Eine PKI ist ein System, das digitale Zertifikate ausstellen, verteilen und prüfen kann.

RA Registration Authority, eine Registrierungsstelle in einer PKI.

RDF Resource Description Framework, ein Modell zum Beschreiben von Aussagen über Ressourcen.

Replay Angriff Ein Angriff, bei dem Daten aufgezeichnet und später erneut verwendet werden.

REST Representational State Transfer, ein Architekturstil für Web Services.

RP Relying Party, ein anderer Begriff für Service Provider in Identitätsmanagementsystemen.

SAML Security Assertion Markup Language, ein Authentifikationsprotokoll, das XML zum Austausch von Authentifizierungs- und Autorisierungsinformationen verwendet.

SCIM System for Cross-Domain Identity Management, ein Datenschema zur Beschreibung von Personen und Gruppen.

Secure Quick Reliable Login Ein Authentifizierungsprotokoll mit QR-Codes.

Service Provider Ein Anbieter von Diensten für Nutzer (z.B. Online-Shopping oder Video-Streaming).

Single-Sign-On Eine Eigenschaft eines System, das Nutzern erlaubt, mehrere Dienste mit nur einer Anmeldung zu benutzen.

SMTP Simple Mail Transfer Protocol, ein Protokoll zum Austausch von E-Mails.

SOAP Simple Object Access Protocol, ein industrieller Standard zum Austausch von Daten.

Social Engineering Ein Angriff, bei dem der Angreifer in persönlichen Kontakt mit dem Opfer tritt, um Informationen zu erhalten. Z.B. durch Anruf als vermeintlicher Techniker.

SP Siehe Service Provider.

SQL-Injection Ein Angriff, bei dem automatisierte Datenbankabfragen über Eingabefelder manipuliert werden.

SQRL Siehe Secure Quick Reliable Login.

SSO Siehe Single-Sign-On.

STS Security Token Service, eine, im WS-Trust-Standard definierte, Komponente des IdP zur Generierung von Tokens.

Symmetrische Verschlüsselung Ein Verschlüsselungssystem, das zum Ver- und Entschlüsseln den selben Schlüssel verwendet.

TGS Ticket-Granting Service, eine Komponente des Identitätsprovider im Kerberos-Protokoll.

TGT Ticket-Granting Ticket, ein spezielles Ticket im Kerberos-Protokoll, das zum Anfordern von weiteren Tickets benötigt wird.

Ticket Ein Token im Kerberos-Protokoll, dass vom TGS erstellt wird, falls das TGT gültig ist.

TLS Transport Layer Security, ein hybrides (symmetrische und asymmetrische Kryptografie) Verschlüsselungsprotokoll zur sicheren Datenübertragung. Nachfolger von SSL (Secure Sockets Layer).

Token Ein physisches oder virtuelles Objekt, das zur Authentifizierung eingesetzt wird. Es es kann signiert oder auch verschlüsselt sein.

Trust Level Eine Wahrscheinlichkeit, die angibt wie sicher ein Endgerät ist, dass der aktuelle Benutzer auch der Besitzer des Geräts ist.

U2F Universal 2nd Factor, ein Zwei-Faktor-Authentizierungsprotokoll von FIDO, dass Wissen und Besitz kombiniert.

UAF Universal Authentication Framework, ein Authentifizierungsprotokoll, das unterschiedliche Authentifikations-Methoden (auch biometrische) erlaubt.

UMA User Managed Access, ein Autorisierungsprotokoll und Erweiterung von OAuth, das dem Nutzer mehr Kontrolle über die Zugriffsrechte auf seine Daten gibt.

URI Uniform Resource Identifier, eine Zeichenfolge zur Identifizierung einer physischen oder abstrakten Ressource.

URL Uniform Resource Locator, eine Zeichenfolge zur Lokalisierung einer Ressource.

VA Validation Authority, ein Validierungsdienst in einer PKI.

Web of Trust Eine PKI, die ohne Root CA auskommt. Jeder kann Zertifikate ausstellen, prüfen und validieren. Das System basiert auf Vertrauen (Trust).

WebID Ein Authentifikationsprotokoll, bei der die Identität durch eine URI und ein RDF-Dokument, das den Nutzer beschreibt, repräsentiert wird.

WS Web Service, ein Internetdienst.

WSDL Web Services Description Language, ein industrieller Standard um Dienste zu beschreiben (z.B. welche Funktionen werden angeboten, welche Datenformate werden verwendet, usw.).

X.509 Ein standardisiertes Format für Zertifikate, das häufig in PKIs eingesetzt wird.

XML Extensible Markup Language, ein strukturiertes und hierarchisches Format für Daten.

Zertifikat Eine Bescheinigung bzw. Urkunde, die Aussagen über eine Entität enthält und beglaubigt ist.

Öffentlicher Schlüssel Ein Teil eines asymmetrischen Verschlüsselungsverfahrens. Damit wird eine Nachricht verschlüsselt oder deren Signatur geprüft.

Literatur

[1] Lujo Bauer, Cristian Bravo-Lillo, Elli Fragkaki, and William Melicher. A comparison of users' perceptions of and willingness to use google, facebook, and google+ single-sign-on functionality. In *Proceedings of the 2013 ACM workshop on Digital identity management*, pages 25–36. ACM, 2013.

[2] Greg E. Blonder. Graphical password, September 24 1996. US Patent 5,559,961.

[3] Don Box, David Ehnebuske, Gopal Kakivaya, Andrew Layman, Noah Mendelsohn, Henrik Frystyk Nielsen, Satish Thatte, and Dave Winer. Simple object access protocol (soap) 1.1, 2000.

[4] Jon Callas, Lutz Donnerhacke, Hal Finney, David Shaw, and Rodney Thayer. Openpgp message format. Technical report, 2007.

[5] Kim Cameron. The laws of identity. *Microsoft Corp*, 2005.

[6] David Chaum. Blind signatures for untraceable payments. In *Advances in cryptology*, pages 199–203. Springer, 1983.

[7] Erik Christensen, Francisco Curbera, Greg Meredith, Sanjiva Weerawarana, et al. Web services description language (wsdl) 1.1, 2001.

[8] OASIS Security Services Technical Committee et al. Security assertion markup language (saml) 2.0, 2012.

[9] Antonella De Angeli, Lynne Coventry, Graham Johnson, and Karen Renaud. Is a picture really worth a thousand words? exploring the feasibility of graphical authentication systems. *International Journal of Human-Computer Studies*, 63(1):128–152, 2005.

[10] Arkajit Dey and Stephen Weis. Pseudoid: Enhancing privacy in federated login. In *Hot Topics in Privacy Enhancing Technologies*, pages 95–107, 2010.

[11] Roy Thomas Fielding. *Architectural styles and the design of network-based software architectures*. PhD thesis, University of California, Irvine, 2000.

[12] Dinei Florencio and Cormac Herley. A large-scale study of web password habits. In *Proceedings of the 16th international conference on World Wide Web*, pages 657–666. ACM, 2007.

[13] OpenID Foundation. OpenID Authentication 2.0 – Final. Technical report, December 2007.

[14] Rachel Greenstadt and Jacob Beal. Cognitive security for personal devices. In *Proceedings of the 1st ACM workshop on Workshop on AISec*, pages 27–30. ACM, 2008.

[15] Kelly Grizzle, Erik Wahlstroem, Chuck Mortimore, and Phil Hunt. System for cross-domain identity management: Core schema. 2015.

[16] Thomas Hardjono. User-managed access (uma) profile of oauth 2.0. *IETF draft*, 27, 2012.

[17] Dick Hardt. The oauth 2.0 authorization framework. 2012.

[18] David Jaeger, Hendrik Graupner, Andrey Sapegin, Feng Cheng, and Christoph Meinel. Gathering and analyzing identity leaks for security awareness. In *Technology and Practice of Passwords*, pages 102–115. Springer, 2015.

[19] Roger M. Needham and Michael D. Schroeder. Using encryption for authentication in large networks of computers. *Communications of the ACM*, 21(12):993–999, 1978.

[20] B. Clifford Neuman and Theodore Ts' O. Kerberos: An authentication service for computer networks. *Communications Magazine, IEEE*, 32(9):33–38, 1994.

[21] David Nuñez and Isaac Agudo. Blindidm: A privacy-preserving approach for identity management as a service. *International Journal of Information Security*, 13(2):199–215, 2014.

[22] Natsuhiko Sakimura, J. Bradley, M. Jones, B. de Medeiros, and C. Mortimore. Openid connect core 1.0. *The OpenID Foundation*, page S3, 2014.

[23] Andrei Sambra, Henry Story, and Tim Berners-Lee. Webid 1.0, web identity and discovery, w3c editor's draft 29 December 2015.

[24] David Solo, Russell Housley, and Warwick Ford. Internet x. 509 public key infrastructure certificate and crl profile. 1999.

[25] San-Tsai Sun, Eric Pospisil, Ildar Muslukhov, Nuray Dindar, Kirstie Hawkey, and Konstantin Beznosov. What makes users refuse web single sign-on?: An empirical investigation of openid. In *Proceedings of the Seventh Symposium on Usable Privacy and Security*, SOUPS '11, pages 4:1–4:20, New York, NY, USA, 2011. ACM.

[26] Xiaoyuan Suo, Ying Zhu, and G. Scott. Owen. Graphical passwords: A survey. In *Proceedings of the 21st Annual Computer Security Applications Conference*, ACSAC '05, pages 463–472, Washington, DC, USA, 2005. IEEE Computer Society.

[27] I. Thomas and C. Meinel. Enhancing claim-based identity management by adding a credibility level to the notion of claims. In *Services Computing, 2009. SCC '09. IEEE International Conference on*, pages 243–250, Sept 2009.

[28] Ivonne Thomas. Soa security and the world of digital identities. `http://www.tele-task.de/de/archive/video/flash/2367`, 2009. besucht am 16.02.2017.

[29] Ivonne Thomas and Christoph Meinel. An identity provider to manage reliable digital identities for soa and the web. In *Proceedings of the 9th Symposium on Identity and Trust on the Internet*, IDTRUST '10, pages 26–36, New York, NY, USA, 2010. ACM.

[30] Ivonne Thomas, Michael Menzel, and Christoph Meinel. Using quantified trust levels to describe authentication requirements in federated identity management. In *Proceedings of the 2008 ACM Workshop on Secure Web Services*, SWS '08, pages 71–80, New York, NY, USA, 2008. ACM.

Aktuelle Technische Berichte des Hasso-Plattner-Instituts

Band	ISBN	Titel	Autoren / Redaktion
113	978-3-86956-394-7	**Blockchain : Technologie, Funktionen, Einsatzbereiche**	Tatiana Gayvoronskaya, Christoph Meinel, Maxim Schnjakin
112	978-3-86956-391-6	**Automatic verification of behavior preservation at the transformation level for relational model transformation**	Johannes Dyck, Holger Giese, Leen Lambers
111	978-3-86956-390-9	**Proceedings of the 10th Ph.D. retreat of the HPI research school on service-oriented systems engineering**	Christoph Meinel, Hasso Plattner, Mathias Weske, Andreas Polze, Robert Hirschfeld, Felix Naumann, Holger Giese, Patrick Baudisch, Tobias Friedrich, Emmanuel Müller
110	978-3-86956-387-9	**Transmorphic : mapping direct manipulation to source code transformations**	Robin Schreiber, Robert Krahn, Daniel H. H. Ingalls, Robert Hirschfeld
109	978-3-86956-386-2	**Software-Fehlerinjektion**	Lena Feinbube, Daniel Richter, Sebastian Gerstenberg, Patrick Siegler, Angelo Haller, Andreas Polze
108	978-3-86956-377-0	**Improving Hosted Continuous Integration Services**	Christopher Weyand, Jonas Chromik, Lennard Wolf, Steffen Kötte, Konstantin Haase, Tim Felgentreff, Jens Lincke, Robert Hirschfeld
107	978-3-86956-373-2	**Extending a dynamic programming language and runtime environment with access control**	Philipp Tessenow, Tim Felgentreff, Gilad Bracha, Robert Hirschfeld
106	978-3-86956-372-5	**On the Operationalization of Graph Queries with Generalized Discrimination Networks**	Thomas Beyhl, Dominique Blouin, Holger Giese, Leen Lambers
105	978-3-86956-360-2	**Proceedings of the Third HPI Cloud Symposium "Operating the Cloud" 2015**	Estee van der Walt, Jan Lindemann, Max Plauth, David Bartok (Hrsg.)
104	978-3-86956-355-8	**Tracing Algorithmic Primitives in RSqueak/VM**	Lars Wassermann, Tim Felgentreff, Tobias Pape, Carl Friedrich Bolz, Robert Hirschfeld

ISBN 978-3-86956-395-4
ISSN 1613-5652