Standards und Schnittstellen für das Identitätsmanagement in der Cloud

Detlef Hühnlein · Johannes Schmölz · Tobias Wich
Benedikt Biallowons · Moritz Horsch · Tina Hühnlein

ecsec GmbH
{vorname.nachname}@ecsec.de

Zusammenfassung

Für ein vertrauenswürdiges Cloud Computing werden zuverlässige Mechanismen für die starke Authentisierung in der Cloud benötigt. Der vorliegende Beitrag liefert einen Überblick über existierende und entstehende Standards und Schnittstellen in diesem Bereich und ordnet diese in die im Rahmen des SkIDentity-Projektes für diesen Zweck entwickelte Referenzarchitektur ein.

1 Einleitung

Dem so genannten „Cloud Computing“ [BYVB+09, NIS11, ShKa09], bei dem verschiedenste IT-Dienste bei Bedarf einfach „aus der Wolke“ bezogen werden können, wird eine große Zukunft vorausgesagt. Beispielsweise soll sich das deutsche Marktvolumen im Bereich der öffentlich angebotenen „Public Clouds“ von 702 Mio. € im Jahr 2010 bis zum Jahr 2025 auf 21,99 Mrd. € erhöhen und somit mehr als verdreißigfachen [Ber10]. Auf der anderen Seite wurde in [SHJS+11] gezeigt, dass selbst die Cloud-Angebote der international führenden Anbieter erfolgreich angegriffen werden können und gezielte Einbrüche in Cloud-Anwendungen [Robe12] zu signifikanten wirtschaftlichen Schäden führen können. Vor diesem Hintergrund ist es wenig verwunderlich, dass das *Bundesamt für Sicherheit in der Informationstechnik* (BSI) in [BSI11b] generell für Administrationszugänge, sowie bei Cloud-Angeboten mit hohem Schutzbedarf, den Einsatz von starken, auf mindestens zwei Faktoren (Besitz, Wissen, Sein, etc.) basierenden, Authentisierungsmechanismen fordert. Damit ein Cloud-Anbieter nicht seinen potentiellen Kunden in einem kostspieligen Prozess erst geeignete Authentisierungstoken zur Verfügung stellen muss, wurde im SkIDentity-Projekt (siehe www.SkIDentity.de und [HHRS+11]) vorgeschlagen, die verschiedenen Chipkarten der eCard-Strategie der Bundesregierung [Kowa07] und weitere bereits im Feld befindliche Authentisierungstoken zusammen mit den entsprechenden Authentisierungsdiensten (z.B. eID-Services[1] für den neuen Personalausweis) für die starke Authentisierung in der Cloud zu nutzen.

Damit die Integration dieser sehr unterschiedlichen Dienste zu einer umfassenden Sicherheitsinfrastruktur für die Cloud gelingen kann, wurden in einer ersten Projektphase die relevanten Standards und Schnittstellen für das sichere Identitätsmanagement in der Cloud identifiziert und in eine umfassende Referenzarchitektur für die starke Authentisierung in der Cloud eingeordnet. Diese „SkIDentity-Referenzarchitektur“ (siehe Abbildung 1 und Abschnitt 2) und die

[1] Siehe http://www.ccepa.de/eid-service-anbieter.

P. Schartner · J. Taeger (Hrsg.) · D•A•CH Security 2012 · syssec (2012) 208-218.

wichtigsten darin integrierten Standards und Schnittstellen (z.B. [CKPM05a, Ope07, JoMc09, ISO08a, CEN08, BSI11a, BSI10, Sun11, Grou97]) sollen in diesem Beitrag kurz vorgestellt werden.

2 Die SkIDentity-Referenzarchitektur

Im Rahmen des SkIDentity-Projektes, das zu den Gewinnern des „Trusted Cloud“[2] Technologiewettbewerbs des *Bundesministerium für Wirtschaft und Technologie* (BMWi) zählt, wurde eine umfassende Referenzarchitektur für die starke Authentisierung in der Cloud entwickelt. Anhand dieser in Abbildung 1 dargestellten Referenzarchitektur werden in Abschnitt 3 die wichtigsten Standards und Schnittstellen für die starke Authentisierung in der Cloud vorgestellt.

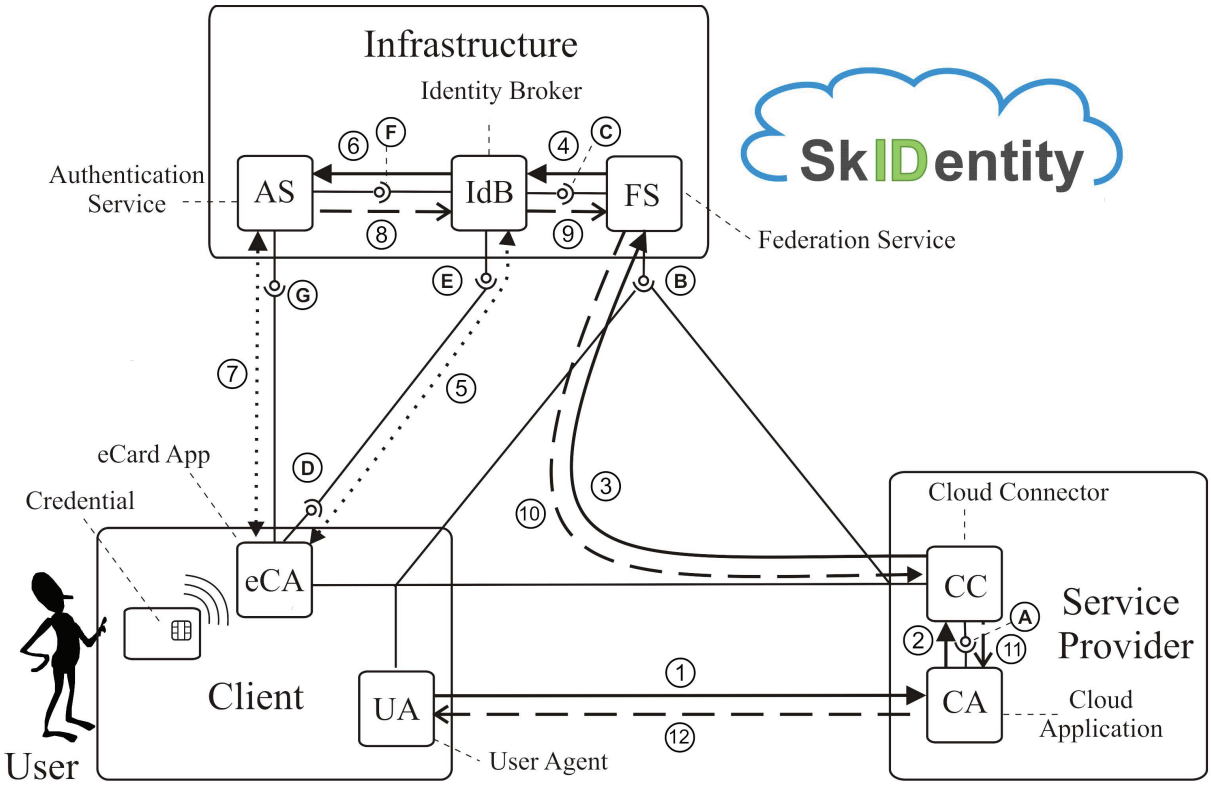

Abb. 1: SkIDentity-Referenzarchitektur für die starke Authentisierung in der Cloud

2.1 Systemkomponenten

Wie in Abbildung 1 ersichtlich, umfasst die SkIDentity-Referenzarchitektur für die starke Authentisierung in der Cloud

- Systemkomponenten beim Benutzer (siehe Abschnitt 2.1.1),
- Systemkomponenten beim Diensteanbieter (siehe Abschnitt 2.1.2), sowie entsprechende
- Infrastrukturkomponenten (siehe Abschnitt 2.1.3).

[2] Siehe www.trusted-cloud.de.

2.1.1 System des Benutzers

Das System des Benutzers (Client) umfasst einen *User Agent* (UA), der beispielsweise durch einen beliebigen Browser realisiert sein kann, und eine so genannte *eCard App* (eCA) (vgl. [BSI12, HPSW+12]), die unter Verwendung des *digitalen Ausweises* (Credential) des *Benutzers* (User) eine Authentisierung gegenüber dem *Authentication Service* (AS) in der Infrastruktur durchführt. Darüber hinaus bietet die eCA eine Schnittstelle, die es dem *Identity Broker* (IdB) ermöglicht, die verfügbaren Credentials und Präferenzen des Benutzers zu ermitteln, so dass ein geeigneter Authentisierungsdienst ausgewählt werden kann.

2.1.2 System des Diensteanbieters

Das System des Diensteanbieters (Service Provider) umfasst die eigentliche Anwendung (*Cloud Application* (CA)) und einen so genannten *Cloud Connector* (CC), der die Kommunikation mit dem *Federation Service* (FS) in der SkIDentity-Infrastruktur übernimmt.

2.1.3 SkIDentity-Infrastruktur

In der SkIDentity-Infrastruktur für die starke Authentisierung in der Cloud existieren *Federation Services* (FS) und *Authentication Services* (AS), die über einen *Identity Broker* (IdB) miteinander verbunden sind. Hierbei führt der AS die tatsächliche Authentisierung durch, während der FS die benötigte Funktionalität für ein möglicherweise gewünschtes Single Sign-On bereitstellt und die hierfür vorgesehenen Föderationsprotokolle unterstützt.

2.2 Wesentliche Schnittstellen

In der SkIDentity-Referenzarchitektur existieren insbesondere die folgenden Schnittstellen, die mit den in Abschnitt 3 vorgestellten Standards realisiert werden können:

(A) *Cloud-Interface* – wird vom CC angeboten und von der CA für die Initiierung des Authentisierungsvorganges genutzt.

(B) *Federation-Interface* – wird vom FS angeboten und vom CC für die Übermittlung einer Authentisierungsanfrage genutzt. Diese Schnittstelle kann durch ein geeignetes Föderationsprotokoll (siehe Abschnitt 3.2) realisert werden.

(C) *Broker-Interface* – wird vom IdB angeboten und vom FS bzw. CC genutzt, um die Authentisierung bei einem angeschlossenen Authentisierungsdienst anzustoßen.

(D) *Credential-Interface* – wird von der eCA angeboten und vom IdB für die Ermittlung der aktuell verfügbaren Credentials sowie der Präferenzen des Benutzers genutzt. Die Schnittstelle orientiert sich am Client-Interface wie es in [BSI11a, Part 7, Section 3.2] definiert ist. Insbesondere wird die eCA hierbei instruiert, über das Dispatcher-Interface (E) des IdB eine XML-Struktur mit weiteren Informationen abzuholen.

(E) *Dispatcher-Interface* – wird vom IdB angeboten und von der eCA für die Ermittlung des für die Transaktion zuständigen Authentisierungsdienstes genutzt. Über diese Schnittstelle wird der eCA eine XML-Struktur bereit gestellt, in der insbesondere die Adresse des für die Transaktion zuständigen Authentisierungsdienstes enthalten ist (vgl. [BSI11a, Part 7, Section 3.3]).

(F) *Authentication-Service-Interface* – wird von den verschiedenen AS angeboten und vom IdB für die Initiierung des Authentisierungsvorganges genutzt. Die detaillierte Ausgestaltung dieses Interfaces hängt von den integrierten Authentisierungsdiensten ab.

(G) *Authentication-Interface* – wird vom AS angeboten und von der eCA für die Durchführung des Authentisierungsprotokolles genutzt. Bei einer Authentisierung mit dem neuen Personalausweis läuft hier beispielsweise das *Extended Access Control* (EAC) Protokoll v2.0 gemäß [BSI08] ab.

2.3 Anwendungsfälle

Die wesentlichen Abläufe bei der Registrierung eines Benutzers an einem Dienst und einer nachfolgenden Anmeldung durch einen Authentisierungsvorgang bzw. in einem Single Sign-On Szenario sollen im Folgenden kurz erläutert werden.

2.3.1 Registrierung eines Benutzers

In diesem Anwendungsfall möchte sich der Benutzer bei der CA registrieren.

(1) $UA \rightarrow CA/CC$: Der Benutzer greift über seinen UA auf eine Ressource zu, die über den CC den Registrierungsprozess initiiert.

(2) CC: Im CC wird daraufhin unter Verwendung der konfigurierten Informationen die Registrierung des Benutzers über den FS angestoßen.

(3) $CC \rightarrow FS$: Über die Schnittstelle (B) und ein geeignetes Föderationsprotokoll werden die für die Registrierung akzeptierten Ausweise oder der geforderte Assurance Level (vgl. [ISO12, NIS06]) sowie eine Liste der benötigten Identitätsattribute an den FS übermittelt. Damit der Benutzer bei einer späteren Authentisierung wieder erkannt werden kann, ist es zweckmäßig, dass die Liste der angefragten Attribute auch einen aus dem Credential des Benutzers extrahierten und bezüglich der Applikation eindeutigen "eIdentifier"[3] enthält.

(4) $FS \rightarrow IdB$: Der FS übergibt wiederum über die Schnittstelle (C) die Registrierungsanforderung an den IdB.

(5) $IdB \leftrightarrow eCA$: Der IdB ermittelt über die Schnittstelle (D) die aktuell an der eCA verfügbaren Credentials sowie etwaige Präferenzen des Benutzers. Auf Basis dieser Informationen und den in Schritt (3) übermittelten Informationen (Assurance Level, gewünschte Attribute etc.) ermittelt der IdB einen geeigneten AS, an den sich die eCA in Schritt (7) wenden kann, um die Authentisierung des Benutzers durchzuführen. Die Adresse dieses Authentisierungsdienstes kann von der eCA über die Schnittstelle (E) beim IdB erfragt werden.

(6) $IdB \rightarrow AS$: In diesem Schritt wird die Registrierungsanfrage über die Schnittstelle (F) an den ausgewählten AS weitergeleitet.

(7) $eCA \leftrightarrow AS$: Die eCA kommuniziert mit dem AS, um die Authentisierung des Benutzers durchzuführen und die gewünschten Attribute aus dem Credential des Benutzers auszulesen.

(8) $AS \rightarrow IdB$: Nach erfolgreicher Authentisierung und dem Ermitteln der angefragten Identitätsattribute liefert der AS diese zurück an den IdB.

(11) $IdB \rightarrow FS$: Der IdB leitet das Ergebnis der Authentisierung und die ermittelten Identitätsattribute unverändert an den FS weiter.

(12) $FS \rightarrow CC$: Der FS bildet daraus eine dem Föderationsprotokoll (siehe Abschnitt 3.2) entsprechende "Assertion", die er an den CC sendet.

[3] Im Fall des neuen Personalausweises würde dieser Identifikator beispielsweise mit dem "Restricted Identification" Protokoll gemäß [BSI08, Part 2, Section 3.5] erzeugt werden.

(13) $CC \rightarrow CA$: Der CC prüft die Assertion und stellt der Cloud Application (CA) die Registrierungsinformationen bereit.

(14) $CA \rightarrow UA$: Das Ergebnis des Registrierungsvorgangs wird dem UA angezeigt.

2.3.2 Authentisierung eines registrierten Benutzers

Der Ablauf bei der Authentisierung eines bereits registrierten Benutzers verläuft analog zur Registrierung, wobei jedoch statt der vollständigen Liste der Identitätsattribute (siehe Schritt (3) in Abschnitt 2.3.1) lediglich der eIdentifier angefordert wird.

2.3.3 Single Sign-On

Sofern für den Zugriff auf die Anwendung nicht zwingend eine aktuelle Authentisierung durchgeführt werden muss, sondern eine bereits zu einem früheren Zeitpunkt erfolgte und im FS vermerkte Authentisierung ausreicht, kann ein "Single Sign-On" [PaMi03] realisiert werden. In diesem Fall beginnt der Ablauf wie in Abschnitt 2.3.1 beschrieben, aber nach Schritt (3) kann der FS sofort mit Schritt (12) fortfahren und die Schritte (4) bis einschließlich (11) würden in diesem Fall durch einen Zugriff auf im FS vorgehaltene Informationen ersetzt werden.

3 Standards und Schnittstellen

In diesem Abschnitt werden die wesentlichen Standards für die starke Authentisierung in der Cloud den verschiedenen Schnittstellen der in Abschnitt 1 dargestellten SkIDentity-Referenzarchitektur zugeordnet.

3.1 (A) – Cloud-Interface

Der CC bietet eine Schnittstelle an, über die die CA den Authentisierungs- und Autorisierungsvorgang anstoßen kann. Statt formaler Standards sind hier vielmehr die "Best Practices" im Bereich Authentisierung und Autorisierung je nach verwendeter Programmiersprache zu berücksichtigen. Beispielsweise bietet sich in der Programmiersprache Java die Verwendung des "Java Authentication and Authorization Service" (JAAS) [Sun11] an, der eine Java-basierte Variante der *Pluggable Authentication Module* (PAM) Architektur [Grou97] darstellt. Hierdurch ist es möglich, den verwendeten Authentisierungsmechanismus – im vorliegenden Fall also insbesondere das genutzte Föderationsprotokoll (siehe Abschnitt 3.2) – gegen ein anderes auszutauschen, ohne dass die Anwendung geändert werden muss.

3.2 (B) – Federation-Interface

Das Federation-Interface für einen bestimmten FS wird durch ein entsprechendes Protokoll für das föderierte Identitätsmanagement realisiert. Hierfür existieren verschiedene Standard-Familien, die in den verschiedenen Anwendungsbereichen (E-Business, E-Government, Social Networks etc.) unterschiedlich weit verbreitet sind (vgl. [HüRZ10]).

3.2.1 SAML

Die vielleicht wichtigste Standard-Familie zur Realisierung des Federation-Interfaces ist mit der von OASIS standardisierten "Security Assertion Markup Language" (SAML) [CKPM05a] gegeben. In diesem aus mehreren Teilen bestehenden Standards wird insbesondere das "Authentication Request Protocol" (siehe [CKPM05a, Section 3.4]) definiert, bei dem durch

Übermittlung einer XML-basierten `AuthnRequest`-Struktur an den FS der Authentisierungsvorgang gestartet werden kann. Das Authentisierungsergebnis wird in einer `Response`-Struktur, die insbesondere eine Folge von `saml:Assertion`-Elementen enthalten kann, zurückgeliefert. Eine solche Assertion kann wiederum neben dem Ergebnis der Authentisierung im `AuthnStatement`-Element auch eine Folge von hierbei ermittelten Attributen in Form eines `AttributeStatement` enthalten. Zu beachten ist, dass beim SAML Standard [CKPM05a] die beiden Vorgänge der Authentisierung mittels `AuthnRequest` und der Ermittlung von Attributen mittels `AttributeQuery` voneinander getrennt sind. Deshalb muss die im eID-Kontext nahe liegende Kombination dieser beiden Funktionen, wie in [AHJM+10] oder [BSI10], abweichend vom Basis-Standard [CKPM05a] leider über das `samlp:Extensions`-Element realisiert werden. Da die Spezifikation der Protokolle [CKPM05a] und Profile [CKPM05b] unabhängig von den darunter liegenden Transportprotokollen ist, können für den jeweiligen Anwendungsfall geeignete Bindings [CHKP+05] genutzt werden. Aus dem Blickwinkel der Sicherheit (siehe z.B. [HiPM05, EiHS09]) ist vor allem das so genannte "Holder-of-Key-Binding" interessant [Klin09, Scav09].

3.2.2 OpenID

Eine leichtgewichtige Alternative für die Realisierung des Federation-Interfaces ist mit dem OpenID-Protokoll [Ope07] gegeben, das zusammen mit existierenden Erweiterungen ausführlich erläutert wird (siehe [EHPS+10]).

3.2.3 WS-*

Das Federation-Interface kann auch auf Basis der bei OASIS standardisierten WS-* Standard-Familie umgesetzt werden. Hierbei wird insbesondere die `wst:RequestSecurityToken` Funktion aus [NGGB+09] in Verbindung mit weiteren Web Service Standards [VOHH+07, NKMH06b, GuHR06, JoMc09]) und darauf aufbauenden Profilen wie [BiP07] genutzt.

3.2.4 OAuth

Wenn neben der Authentisierung auch die Autorisierung in einem verteilten Cloud-System erfolgen soll, bietet sich für die Realisierung des Federation-Interfaces der REST-basierte OAuth 1.0 Standard [Hamm10] oder zukünftig die derzeit von der IETF standardisierte OAuth 2.0 Standard-Familie (siehe http://datatracker.ietf.org/wg/oauth/) an. Diese Schnittstellen werden bislang insbesondere im Umfeld sozialer Netzwerke (z.B. bei Facebook, LinkedIn und XING) eingesetzt und könnten deshalb langfristig möglicherweise eine große Verbreitung erreichen.

3.3 (C) – Broker-Interface

Das Broker-Interface dient dazu, die unterschiedlichen Authentisierungsdienste über eine einheitliche Schnittstelle zugänglich zu machen. Welche Schnittstelle besonders geeignet ist, hängt im Allgemeinen von den Schnittstellen der zu unterstützenden Authentisierungsdienste (siehe Abschnitt 3.6) ab. Auf Basis der vorher existierenden SAML-Profile [AHJM+10, BSI10] wurde mit der SOAP-basierten `Authenticate`-Funktion in [CEN08, Part 3, Chapter 11] eine besonders flexible[4] Schnittstelle für diesen Zweck spezifiziert. Sofern kein eigenständiger FS

[4] Mit dieser Schnittstelle können beliebige Authentisierungspolicies gemäß [VOHH+07] und wie bei SAML [CKPM05a] beliebige Attribute verarbeitet werden.

genutzt werden soll, kann der CC das Broker-Interface auch direkt über eine entsprechend abgesicherte Verbindung aufrufen und gleichzeitig den UU zum IdB umleiten, wodurch das so genannte „Simple Federation Protocol“ (SFP) entsteht.

3.4 (D) – Credential-Interface

Über das Credential-Interface kann der IdB die Fähigkeiten der eCA sowie die Präferenzen des Benutzers abrufen, die in der eCA hinterlegt sind. Diese Informationen sind notwendig, damit der IdB entscheiden kann zu welchem Authentication Service er den Benutzer umleiten muss. Die Schnittstelle orientiert sich am Client-Interface wie es in [BSI11a, Part 7, Section 3.2] definiert ist. Insbesondere wird die eCA hierbei instruiert, über das Dispatcher-Interface (E) des IdB eine XML-Struktur mit weiteren Informationen abzuholen.

3.5 (E) – Dispatcher-Interface

Das vom IdB angebotene Dispatcher-Interface stellt der eCA eine XML-Struktur (vgl. [BSI11a, Part 7, Section 3.3]) bereit, in der insbesondere die Adresse des für die Transaktion zuständigen Authentisierungsdienstes enthalten ist.

3.6 (F) – Authentication-Service-Interface

Über das Authentication-Service-Interface kann der IdB eine Authentisierungsanfrage an den AS stellen. Da das Authentication-Service-Interface nicht für jeden Authentisierungsdienst identitsch ist, muss der IdB eine Vielzahl von unterschiedlichen Schnittstellen bedienen können.

Für die Nutzung des neuen Personalausweises (nPA) spielt die SOAP-basierte eID-Schnittstelle gemäß [BSI10, Kapitel 4] eine wichtige Rolle, da über diese eine Authentisierung mit dem nPA bei einem eID-Server angestoßen werden kann. Darüber hinaus ist in [BSI10, Anhang A] eine SAML-basierte Variante der eID-Schnittstelle definiert, die alternativ genutzt werden kann. Um die Nutzung dieser SAML-basierten Schnittstelle zu erleichtern, werden typischer Weise entsprechende „eID-Connector“ Komponenten bereitgestellt. Außerdem können Authentisierungsdienste über die oben genannte `Authenticate`-Schnittstelle gemäß [CEN08, Part 3, Chapter 11], das PEPS-Interface [AHJM+10] der STORK Initiative, die Active Directory Services [Mic05] oder das von der Firma Reiner SCT initiierte OWOK [Rei12] genutzt werden.

3.7 (G) – Authentication-Interface

Für die Authentifizierung von Benutzern, Geräten und Diensten existieren zahllose kryptographische Protokolle [BoMa03, MePV97]. Selbst bei einer sehr abstrakten Betrachtungsweise können verschiedenste Standardmechanismen unterschieden werden [ISO/09, ISO/08b]. Noch komplexer wird das Bild, wenn zusätzlich die technische Einbettung dieser Verfahren in Kommunikationsprotokolle auf den verschiedenen Schichten des OSI-Modells berücksichtigt wird. Für die Realisierung einer starken Authentisierung in der Cloud erscheinen hierbei beispielsweise die folgenden Standards und Authentisierungsprotokolle relevant:

- Transport Layer Security (TLS) Protokoll [DiRe08],
- Web Service Security basierte Mechanismen zur Authentisierung (siehe [NKMH06b, NKMH06a, NKMH06c])
- Simple Authentication and Security Layer (SASL) basierte Mechanismen in [MeZe06][5]

[5] Siehe auch http://www.iana.org/assignments/sasl-mechanisms/sasl-mechanisms.xml.

- Extended Access Control (EAC) Protokoll [BSI08] über die in [BSI11a] spezifizierten XML-basierten Nachrichtenformate und weitere in [ISO08a, Part 3 Annex A, Part 6] spezifizierte Authentisierungsprotokolle für eID-Karten und schließlich
- OTP-basierte Authentisierungsprotokolle der oath-Initiative[6] [MBHN+05, MBHN+11, MRBM+11]

4 Zusammenfassung

Die bekannt gewordenen Angriffe gegen prominente Cloud Angebote [SHJS+11, Robe12] unterstreichen die Bedeutung der starken Authentisierung von privilegierten Nutzern in der Cloud, wie sie vom BSI in [BSI11b] gefordert wird. Vor diesem Hintergrund wurden im SkIDentity-Projekt (siehe www.SkIDentity.de) die hierfür maßgeblichen internationalen Standards identifiziert und zu einer umfassenden Referenzarchitektur (siehe Abbildung 1) für die starke Authentisierung in der Cloud integriert. Mit Unterstützung des Bundesministeriums für Wirtschaft und Technologie (BMWi) wird diese Referenzarchitektur sukzessive umgesetzt und kann fortan von interessierten Parteien für die starke Authentisierung in der Cloud genutzt werden.

Literatur

[AHJM+10] J. Alcalde-Morano, J. L. Hernández-Ardieta, A. Johnston, D. Martinez, B. Zwattendorfer, M. Stern, J. Heppe: Interface Specification. STORK Deliverable D5.8.2b, 04.10.2010 (2010).

[Ber10] Berlecon Research & al.: Das wirtschaftliche Potenzial des Internet der Dienste. Studie im Auftrag des Bundesministeriums für Wirtschaft und Technologie (BMWi) (2010), http://www.berlecon.de/idd.

[BiP07] BiPRO e.V.: Norm 410 – Security Token Service. BiPRO Norm, Release 1, Version 1.0, vom 19. Juni 2007 (2007), http://docs.ecsec.de/BiPRO-410.

[BoMa03] C. Boyd, A. Mathuria: Protocols for authentication and key establishment (2003).

[BSI08] BSI: Advanced Security Mechanism for Machine Readable Travel Documents - Extended Access Control (EAC). Technical Directive (BSI-TR-03110), Version 2.0 - Release Candidate (2008).

[BSI10] BSI: eID-Server. Technical Directive (BSI-TR-031030), Version 1.4, 14.09.2010 (2010), https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR03130/TR-03130_TR-eID-Server_V1_4_pdf.pdf?__blob=publicationFile.

[BSI11a] BSI: eCard-API-Framework. Technical Directive (BSI-TR-03112), Version 1.1.1, Part 1-7 (2011), https://www.bsi.bund.de/ContentBSI/Publikationen/TechnischeRichtlinien/tr03112/index_htm.html.

[BSI11b] BSI: Sicherheitsempfehlungen für Cloud Computing Anbieter – Mindestsicherheitsanforderungen in der Informationssicherheit. Eckpunktepapier (2011), http://docs.ecsec.de/BSI-MSACC.

[BSI12] BSI: Offizielles Portal für die „AusweisApp“ (2012), http://www.ausweisapp.de.

[6] Siehe http://www.openauthentication.org/.

[BYVB+09] R. Buyya, C. Yeo, S. Venugopal, J. Broberg, I. Brandic: Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. In: *Future Generation Computer Systems*, 25, 6 (2009), 599–616, http://www.buyya.com/gridbus/papers/Cloud-FGCS2009.pdf.

[CEN08] Comité Européen de Normalisation (CEN): Identification card systems – European Citizen Card – Part 1-4. (Draft of) Technical Specification (2008).

[CHKP+05] S. Cantor, F. Hirsch, J. Kemp, R. Philpott, E. Maler: Bindings for the OASIS Security Assertion Markup Language (SAML) V2.0. OASIS Standard, 15.03.2005 (2005), http://docs.oasisopen.org/security/saml/v2.0/saml-bindings-2.0-os.pdf.

[CKPM05a] S. Cantor, J. Kemp, R. Philpott, E. Maler: Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) V2.0. OASIS Standard, 15.03.2005 (2005), http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf.

[CKPM05b] S. Cantor, J. Kemp, R. Philpott, E. Maler: Profiles fo the OASIS Security Assertion Markup Language (SAML) V2.0. OASIS Standard, 15.03.2005 (2005), http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf.

[DiRe08] T. Dierks, E. Rescorla: The Transport Layer Security (TLS) Protocol Version 1.2. Request For Comments – RFC 5246 (2008), http://www.ietf.org/rfc/rfc5246.txt.

[EHPS+10] D. Eske, D. Hühnlein, S. Paulus, J. Schmölz, T. Wich, T. Wieland: OpeneGK – Benutzerfreundliche und sichere Authentisierung für Mehrwertdienste im Gesundheitswesen. *In: Tagungsband* perspeGKtive 2010, GI-Edition (2010), *LNI*, Bd. 174, 83–103, http://www.ecsec.de/pub/openeGK.pdf.

[EiHS09] J. Eichholz, D. Hühnlein, J. Schwenk: SAMLizing the European Citizen Card. *In: Proceedings of* BIOSIG 2009: Biometrics and Electronic Signatures, GI-Edition (2009), *Lecture Notes in Informatics (LNI)*, Bd. 155, 105–117, http://www.ecsec.de/pub/SAMLizing-ECC.pdf.

[Grou97] T. O. Group: X/Open Single Sign-on Service (XSSO) - Pluggable Authentication Modules. X/Open Document Number: P702, Preliminary Specification (1997), http://www.opengroup.org/onlinepubs/008329799/toc.htm.

[GuHR06] M. Gudgin, M. Hadley, T. Rogers: Web Services Addressing 1.0 - Core. W3C Recommendation (2006), http://www.w3.org/TR/ws-addr-core.

[Hamm10] E. Hammer-Lahav: The OAuth 1.0 Protocol. Request For Comments – RFC 5849 (2010), http://www.ietf.org/rfc/rfc5849.txt.

[HHRS+11] D. Hühnlein, G. Hornung, H. Rossnagel, J. Schmölz, T. Wich, J. Zibuschka: SkIDentity - Vertrauenswürdige Identitäten für die Cloud. DACH-Security 2011 (2011).

[HiPM05] F. Hirsch, R. Philpott, E. Maler: Security and Privacy Considerations for the OASIS Security Assertion Markup Language (SAML) V2.0. OASIS Standard, 15.03.2005 (2005), http://docs.oasis-open.org/security/saml/v2.0/saml-sec-consider-2.0-os.pdf.

[HPSW+12] D. Hühnlein, D. Petrautzki, J. Schmölz, T. Wich, M. Horsch, T. Wieland, J. Eichholz, A. Wiesmaier, J. Braun, F. Feldmann, S. Potzernheim, J. Schwenk,

C. Kahlo, A. Kühne, H. Veit: On the design and implementation of the Open eCard App. *In: Sicherheit 2012*, GI-LNI (2012).

[HüRZ10] D. Hühnlein, H. Rossnagel, J. Zibuschka: Diffusion of Federated Identity Management. to appear (2010).

[ISO08a] ISO/IEC 24727: Identification cards – Integrated circuit cards programming interfaces – Part 1-6 (2008).

[ISO/08b] ISO/IEC: ISO/IEC 11770: Information Technology – Security Techniques – Key Management – Parts 1-4. International Standards (1996-2008).

[ISO/09] ISO/IEC: ISO/IEC 9798: Information Technology – Security Techniques – Entity Authentication – Part 1-6. International Standards (1997-2009).

[ISO12] ISO/IEC DIS 29115: Information technology – Security techniques – Entity authentication assurance framework. International Standard (2012).

[JoMc09] M. B. Jones, M. McIntosh: Identity Metasystem Interoperability Version 1.0. OASIS Standard (2009), http://docs.oasis-open.org/imi/identity/v1.0/os/identity-1.0-spec-os.pdf.

[Klin09] N. Klingenstein: SAML V2.0 Holder-of-Key Web Browser SSO Profile. OASIS Committee Draft 02, 05.07.2009 (2009), http://www.oasis-open.org/committees/download.php/33239/sstc-saml-holder-of-key-browser-sso-cd-02.pdf.

[Kowa07] B. Kowalski: Die eCard-Strategie der Bundesregierung im Überblick. *In: BIOSIG 2007: Biometrics and Electronic Signatures* (2007), *LNI*, Bd. 108, 87–96, http://subs.emis.de/LNI/Proceedings/Proceedings108/gi-proc-108-008.pdf.

[MBHN+05] D. M'Raihi, M. Bellare, F. Hoornaert, D. Naccache, O. Ranen: HOTP: An HMAC-Based One-Time Password Algorithm. Request For Comments – RFC 4226 (2005), http://www.ietf.org/rfc/rfc4226.txt.

[MBHN+11] D. M'Raihi, M. Bellare, F. Hoornaert, D. Naccache, O. Ranen: TOTP: Time-Based One-Time Password Algorithm. Request For Comments – RFC 6238 (2011), http://www.ietf.org/rfc/rfc6238.txt.

[MePV97] A. J. Menezes, Paul C. van Oorschot, S. A. Vanstone: Handbook of Applied Cryptography. CRC Press (1997), http://www.cacr.math.uwaterloo.ca/hac/.

[MeZe06] A. Melnikov, K. Zeilenga: Simple Authentication and Security Layer (SASL). Request For Comments – RFC 4422 (2006), http://www.ietf.org/rfc/rfc4422.txt.

[Mic05] Microsoft Inc.: Active Directory (2005), http://www.microsoft.com/windows server2008/en/us/active-directory.aspx.

[MRBM+11] D. M'Raihi, J. Rydell, S. Bajaj, S. Machani, D. Naccache: OCRA: OATH Challenge-Response Algorithm. Request For Comments – RFC 6287 (2011), http://www.ietf.org/rfc/rfc6287.txt.

[NGGB+09] A. Nadalin, M. Goodner, M. Gudgin, A. Barbir, H. Granqvist: WS-Trust 1.4. OASIS Standard, 02.02.2009 (2009), http://docs.oasis-open.org/ws-sx/ws-trust/v1.4/os/ws-trust-1.4-spec-os.pdf.

[NIS06] NIST: Electronic Authentication Guideline. NIST Special Publication 800-63 Version 1.0.2 (2006), http://csrc.nist.gov/publications/nistpubs/800-63/SP800-63V1_0_2.pdf.

[NIS11] NIST: The NIST Definition of Cloud Computing. NIST Special Publication 800-145 (2011), http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf.

[NKMH06a] A. Nadalin, C. Kaler, R. Monzillo, P. Hallam-Baker: Web Services Security Kerberos Token Profile 1.1. OASIS Standard, 01.02.2006 (2006), http://www.oasis-open.org/committees/download.php/16788/wss-v1.1-spec-os-KerberosTokenProfile.pdf.

[NKMH06b] A. Nadalin, C. Kaler, R. Monzillo, P. Hallam-Baker: Web Services Security: SOAP Message Security 1.1. OASIS Standard, 01.02.2006 (2006), http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf.

[NKMH06c] A. Nadalin, C. Kaler, R. Monzillo, P. Hallam-Baker: Web Services Security X.509 Certificate Token Profile 1.1. OASIS Standard, 01.02.2006 (2006), http://www.oasis-open.org/committees/download.php/16785/wss-v1.1-spec-os-x509TokenProfile.pdf.

[Ope07] OpenID Foundation: OpenID Authentication 2.0. Final, December 5, 2007 (2007), http://openid.net/specs/openid-authentication-2_0.html.

[PaMi03] A. Pashalidis, C. Mitchell: A taxonomy of single sign-on systems. *In: Information Security and Privacy*, Springer (2003), 219–219.

[Rei12] Reiner SCT: OWOK – One Web, One Key (2012), http://www.reiner-sct.com/owok/.

[Robe12] P. Roberts: Cloud Service Linode Hacked, Bitcoin Accounts Emptied. Threat Post, 02.03.2012 (2012), http://docs.ecsec.de/Robe12.

[Scav09] T. Scavo: SAML V2.0 Holder-of-Key Assertion Profile. OASIS Committee Draft 02, 05.07.2009 (2009), http://www.oasis-open.org/committees/download.php/33236/sstc-saml2-holder-of-key-cd-02.pdf.

[SHJS+11] J. Somorovsky, M. Heiderich, M. Jensen, J. Schwenk, N. Gruschka, L. L. Iacono: All your clouds are belong to us: security analysis of cloud management interfaces. *In: C. Cachin, T. Ristenpart (Hrsg.), CCSW*, ACM (2011), 3–14.

[ShKa09] S. Shankland, J. Kaden: Gartner: Cloud Computing wird wichtigster IT-Trend 2010. ZDNet-Beitrag, 21.10.2009 (2009), http://docs.ecsec.de/ShKa09.

[Sun11] Sun Inc.: Java Authentication and Authorization Service (JAAS). Reference Guide for the Java TM SE Development Kit 6 (2011), http://java.sun.com/javase/6/docs/technotes/guides/security/jaas/JAASRefGuide.html.

[VOHH+07] A. S. Vedamuthu, D. Orchard, F. Hirsch, M. Hondo, P. Yendluri, T. Boubez, Ümit Yalcinalp: Web Services Policy 1.5 - Framework (WS-Policy). W3C Recommendation (2007), http://www.w3.org/TR/ws-policy.