OpeneGK – Benutzerfreundliche und sichere Authentisierung für Mehrwertdienste im Gesundheitswesen

Daniel Eske[1] · Detlef Hühnlein[1] · Sachar Paulus[2]
Johannes Schmölz[1,3] · Tobias Wich[1,3] · Thomas Wieland[3]

[1] ecsec GmbH, Sudetenstr. 16, 96247 Michelau,
{daniel.eske,detlef.huehnlein,johannes.schmoelz,tobias.wich}@ecsec.de

[2] paulus.consult, Am Mühlrain 21/2, 69151 Neckargemünd
sachar.paulus@paulus-consult.de

[3] Hochschule Coburg, Friedrich-Streib-Str. 2, 96450 Coburg
thomas.wieland@hs-coburg.de

Abstract: Dieser Beitrag zeigt, wie die elektronische Gesundheitskarte (eGK) in Verbindung mit dem OpenID-Protokoll bei web-basierten Mehrwertdiensten im Gesundheitswesen zur sicheren, datenschutz- und benutzerfreundlichen Registrierung und Authentisierung genutzt werden kann. Außerdem verspricht die Kombination mit dem weit verbreiteten OpenID-Protokoll eine schnellere Akzeptanz und Verbreitung der eGK-basierten Authentisierung im Internet.

1 Einleitung

Im Gesundheitswesen gibt es eine steigende Zahl von Web-basierten Diensten. Waren diese herkömmlicherweise eher Informationssammlungen und Nachschlagewerke, die zum Teil einfache Foren beinhalteten, so finden sich inzwischen mehr und mehr Lösungen, die stärker personalisierte Inhalte anbieten und sehr individuell auf die Bedürfnisse der einzelnen Nutzer eingehen – sei es zur gegenseitigen Selbsthilfe oder zur gezielten professionellen Beratung. Einige Anbieter machen bereits das Anlegen von elektronischen Gesundheitsakten im Internet möglich[1].

Anders als bei den meisten Anwendungen im Internet, wie etwa sozialen Netzwerken, geht es bei Web-Angeboten im Gesundheitsweisen um besonders sensible Informationen, nämlich Informationen über den Gesundheitszustand des Nutzers. Daher werden hohe Anforderungen an den Datenschutz an derartige Anwendungen gestellt: Jeder Nutzer muss allein darüber entscheiden können, an wen er welche Auskünfte weitergibt, und er muss sich zudem sicher sein können, dass diese Angaben nicht missbraucht oder weitergegeben werden. Diese Anforderungen bedeuten für die Entwicklung wie auch den operativen Betrieb eines Web-Dienstes, dass ein besonders hohes Maß an Sicherheit erreicht werden muss.

[1] Siehe z.B. http://www.gesundheitsakte.de, http://www.onmeda.de/ratgeber/gesundheitswesen/gesundheitsakte.html, https://www.lifesensor.com/de/de/ oder http://www.econmed.de/produkte/gesundheitsakte.html

Leider sind viele Anwendungen im Internet nicht in der Lage, diese Anforderungen zu erfüllen. Dies beginnt oft bereits bei der Authentisierung, die meist ausschließlich über Benutzername und Passwort abgewickelt wird. Der Anbieter kann sich nicht sicher sein, dass der Benutzer auch derjenige ist, der er vorgibt zu sein. Gleichzeitig hat der Nutzer keine Garantie, dass nicht Unbefugte, etwa durch Ausspähen (Phishing, Pharming) oder Erraten des Passworts, Zugriff auf seine Daten erlangen.

Damit personalisierte medizinische Mehrwertdienste im Internet Akzeptanz in der Bevölkerung finden, muss ein hoher Sicherheitsstandard angesetzt werden, gleichzeitig aber auch ein hoher Bedienungskomfort sichergestellt werden. Eine Möglichkeit dazu bietet die starke Authentisierung mit einem physikalischen kryptographischen Sicherheitstoken, z.B. einer Chipkarte. Dann benötigt ein Angreifer neben dem Wissen auch noch den Besitz des Authentizitätsnachweises, was das Risiko deutlich senken würde. Eine solche Chipkarte müsste aber den Benutzern in einfacher und weit verbreiteter Form zugänglich gemacht und genutzt werden, um bei Anbietern wie bei Benutzern wirklich akzeptiert und damit marktfähig zu werden. Spezialisierte, Dienste-spezifische Kartenlösungen einzelner Anbieter[2] können dies voraussichtlich nicht erreichen.

Die Deutsche elektronische Gesundheitskarte erfüllt genau diese Anforderungen. Sie soll an alle gesetzlich Versicherten ausgegeben werden, unterstützt eine starke Authentisierung und stellt damit im Prinzip ein attraktives Authentisierungswerkzeug für Web-basierte Anwendungen im Gesundheitswesen dar.

Allerdings sind die an die eGK geknüpften Vorgaben und Sicherheitsanforderungen vergleichsweise komplex, so dass vermutlich nur wenige Anbieter den Aufwand investieren werden, eine starke Authentisierung auf Basis der eGK innerhalb ihrer Anwendung zu realisieren. Wir schlagen daher eine Variante des OpenID-Protokolls[3] vor, in der die eGK zur sicheren, datenschutz- und benutzerfreundlichen Registrierung und Authentisierung genutzt wird. Hierdurch wird der komplexe Zugriff auf die eGK an den zentralen und speziell gesicherten OpeneGK-Dienst delegiert, den der einzelne Mehrwertanbieter nur noch für sich nutzbar machen muss. Für den Anwender ergibt sich damit eine einfache Registrierung bei einem neuen Web-basierten Mehrwertdienst und ein benutzerfreundliches Single Sign-On (SSO).

Durch die Verwendung des OpenID-Protokolls ist bereits vor dem Rollout der eGK eine (schwache) Authentisierung gegenüber dem OpeneGK-Dienst oder einem anderen OpenID-Provider möglich; nach dem flächendeckenden Rollout der elektronischen Gesundheitskarte kann das System dann für verschieden starke Stufen der Authentisierung genutzt werden. Damit kann für jeden Anwendungsfall die optimale Mischung aus Sicherheit und Benutzerfreundlichkeit gewählt werden.

Im Folgenden wollen wir zunächst (vgl. Abschnitt 2) das Konzept von OpenID vorstellen und einige relevante technische Aspekte der eGK hervorheben. In Abschnitt 3 stellen wir dann die Idee von OpeneGK vor und gehen näher auf die beteiligten Systemkomponenten (Bürgerclient, Mehrwertdienst und OpeneGK-Dienst) ein. Schließlich diskutieren wir in Abschnitt 4 die aus diesem Ansatz erwachsenden Konsequenzen und geben in Abschnitt

[2] Siehe beispielsweise `http://www.vita-x.de`.
[3] Siehe `http://openid.net/` und [Raep09].

5 einen Ausblick auf mögliche zukünftige Entwicklungen.

2 Grundlagen

In diesem Abschnitt werden die in diesem Papier benötigten Grundlagen zusammengetragen. Hierbei geht Abschnitt 2.1 auf das OpenID-Protokoll und Abschnitt 2.2 auf die relevanten Aspekte der elektronischen Gesundheitskarte ein.

2.1 OpenID

OpenID wurde ursprünglich entwickelt, um einen einfachen Login-Mechanismus für LiveJournal[4]-basierte Weblogs zu realisieren und zählt heute neben der Security Assertion Markup Language (SAML) [SAML(v2.0)] und dem Identity Metasystem Interoperability Profile [ID-MI(v1.0)] zu den vielversprechendsten Ansätzen für das Web-basierte Single Sign-On, die derzeit im Rahmen der Kantara-Initiative[5] harmonisiert werden sollen.

Beispielsweise wird OpenID von Google, Yahoo, MySpace und AOL, sowie etlichen weiteren namhaften Organisationen unterstützt [OpenID-Pro]. Seit kurzem zählt beispielsweise auch NTT docomo, der größte japanische Mobilfunkanbieter mit mehr als 55 Millionen Kunden [Saki10] zu den Unterstützern von OpenID. Bereits im Jahr 2009 wurde die Grenze von einer Milliarde OpenID-Benutzerkonten überschritten [Kiss09, Wagn09]. Selbst wenn vermutlich nicht all diese Konten aktiv genutzt werden, so verdeutlicht diese Zahl das große Potenzial von OpenID. Frühere Versuche Single Sign-On-Lösungen im Internet zu etablieren scheiterten aus verschiedenen Gründen oder konnten sich nur in Teilgebieten durchsetzen. So fand das im Jahr 1999 von Microsoft eingeführte .NET Passport aufgrund der zentralen Speicherung der Profildaten in Zusammenhang mit der Monopolstellung Microsofts und dessen Lizenzpolitik nur geringe Akzeptanz unter Anbietern und Anwendern und wurde schließlich zur „Windows Live ID“ weiter entwickelt, die inzwischen auch OpenID[6] unterstützt.

Die von der Liberty Alliance[7] erarbeiteten Konzepte für das Single Sign-On erreichten in der Praxis nur eine vergleichsweise geringe Verbreitung, sie bildeten aber den Grundstein für die Entwicklung und Standardisierung von SAML. Während sich SAML im Bereich der öffentlichen Verwaltung (E-Government) und im geschäftlichen Umfeld (E-Business) aufgrund der Möglichkeit, sehr spezifische Policies zu unterstützen, langsam verbreitet [HRZ10], steht einer schnellen Akzeptanz im Consumer-Bereich insbesondere die vergleichsweise große Komplexität von SAML entgegen. Während SAML erst noch entsprechende Profile für den typischen Anwendungsfall, bei dem man eine Authentisierung mit

[4] Siehe `http://www.livejournal.com/`.
[5] Siehe `http://kantarainitiative.org`.
[6] Siehe `http://winliveid.spaces.live.com/Blog/cns!AEE1BB0D86E23AAC!1745.entry`.
[7] Siehe `http://www.projectliberty.org/`.

einer bestimmten Qualitätsstufe und einige Attribute des Benutzers anfordert, erfordert [EHS10], unterstützen die einfach gestalteten OpenID-Spezifikationen und zahlreichen frei verfügbaren Implementierungen den üblichen Consumer-Anwendungsfall bereits seit geraumer Zeit und es ist deshalb zu erwarten, dass sich OpenID für die Authentisierung im Internet weiter verbreiten und möglicher Weise auch durchsetzen wird [HRZ10]. SAML hingegen wird bei B2B- und B2A-Prozessen, bei denen komplexer Policies und die Weitergabe von Berechtigungsinformationen erforderlich sind, vermutlich erste Wahl bleiben.

Wie in Abbildung 1 ersichtlich, besteht das OpenID-System aus einem User (U), der mit seinem User Agent (UA) (z.B. seinem Web-Browser) und mit Unterstützung des OpenID-Providers (OP) auf einen von der Relying Party (RP) angebotenen Dienst zugreifen möchte.

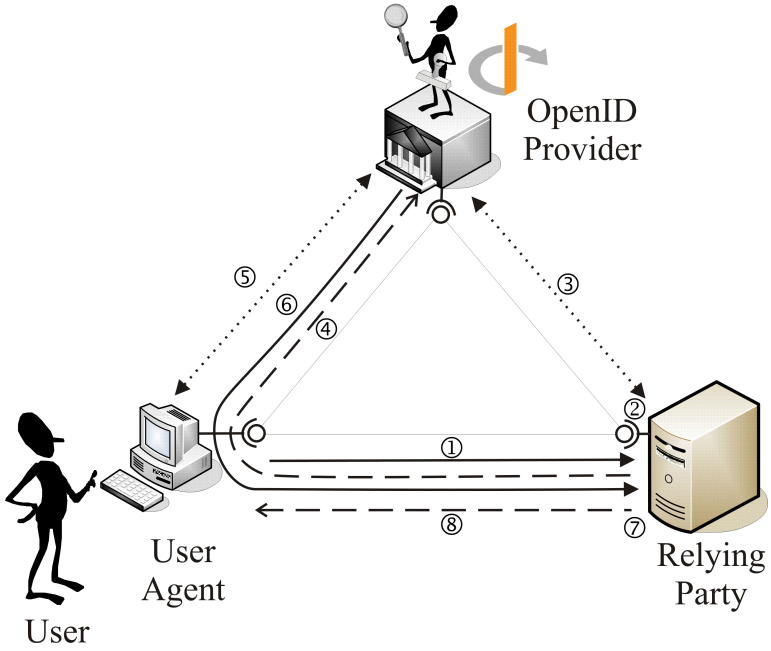

Abbildung 1: Authentisierung und Registrierung mit OpenID

Bevor der User den von der Relying Party angebotenen Dienst nutzen kann sind folgende Schritte nötig (vgl. [OpenID-Auth(v2.0), Section 3]):

1. $UA \rightarrow RP$: Der UA möchte auf eine Ressource zugreifen und kontaktiert daher die RP, die diese Ressource anbietet.
2. RP: Die RP löst die User ID mittels Yadis [Yadis(v1.0)], XRI und XRDS oder einer anderen geeigneten Discovery-Methode auf [RCT08]. Im einfachsten Fall ist hier aber nichts zu tun, da der Benutzer seinen OpenID-Identifier in Form einer URL bereitstellen kann.

3. $RP \rightarrow OP$ (optional): In diesem optionalen Schritt kann die RP (z.B. durch eine Diffie-Hellman-Schlüsselvereinbarung) mit dem OP einen geheimen Schlüssel $K_{RP,OP}$ vereinbaren, der später zur Prüfung der Authentizität des vom OP ausgestellten Authentisierungstoken $T = MAC(m, K_{RP,OP})$ genutzt wird. Sofern bereits eine entsprechende Sicherheitsbeziehung vorhanden ist oder – was aus Sicherheitsgründen *nicht* empfehlenswert ist [Lind09] – die Prüfung des Authentisierungstoken in Schritt 7 an den OP delegiert werden soll, kann dieser Schritt entfallen.

4. $RP \rightarrow OP$: Die RP leitet den UA zum OP um.

5. $OP \leftrightarrow U(A)$: Der UA muss sich am OP in geeigneter Weise authentisieren.

6. $OP \rightarrow RP$: Nach erfolgreicher Authentisierung erzeugt der OP ein entsprechendes Authentisierungstoken T, das durch eine Umleitung des UA zum RP gelangt.

7. RP: Die RP prüft das in der Nachricht enthaltene Authentisierungstoken $T = MAC(m, K_{RP,OP})$. Sofern noch keine Sicherheitsbeziehung zwischen RP und OP (vgl. Schritt 3) existiert und somit der Schlüssel $K_{RP,OP}$ der RP gar nicht bekannt ist, kann die RP die Prüfung von T an den OP delegieren. Wie in [Lind09] erläutert, sollte diese Variante aber aus offensichtlichen Sicherheitsgründen nicht genutzt werden.

8. $RP \rightarrow UA$: Der UA erhält schließlich Zugriff auf die gewünschte Ressource.

Erweiterungen. Für das grundlegende OpenID-Protokoll existieren derzeit die folgenden Erweiterungen:

- *Simple Registration Extension (SRE)*
  Die Simple Registration Extension [OpenID-SRE(v1.0)] bietet einen sehr „leichtgewichtigen“Austausch von Profildaten eines OP-Users zu einem RP. Die Intention der Erweiterung ist ein einfacher und schneller Datenaustausch von insgesamt neun gebräuchlichen Profilattributen (`nickname`, `fullname`, `email` etc.) für die Erstellung eines Benutzerkontos bei einer RP.

- *Provider Authentication Policy Extension (PAPE)*
  Mit der Provider Authentication Policy Extension [OpenID-PAPE(v1.0)] kann eine RP festlegen, wie sich ein Nutzer bei seinem OpenID-Provider zu authentisieren hat. Hierbei kann festgelegt werden, wie lange die Authentisierung maximal zurückliegen darf (`max_auth_age`), welche grundsätzliche Policy[8] vom OpenID-Provider zur Authentisierung genutzt werden soll (`preferred_auth_policies`) und ggf. welcher „Assurance Level“ (siehe z.B. [NIST-800-63]) mit der Authentisierung erreicht werden soll.

[8] Neben den drei in [OpenID-PAPE(v1.0)] definierten Authentication Policies (d.h. `phishing-resistant`, `multi-factor`, `multi-factor-physical`) können eigene Policies definiert werden (vgl. Abschnitt 3.4).

- *Attribute Exchange Erweiterung (AX)*
  Die Attribute Exchange Erweiterung [OpenID-AX(v1.0)] dient zum Austausch von Attributen zwischen RP und OP. Anders als bei der oben erläuterten Simple Registration Erweiterung können hier beliebige Attribute mit der `Fetch`-Operation beim OP gelesen und mit der `Store`-Operation gespeichert werden. Auch wenn die Spezifikation [OpenID-AX(v1.0)] hierzu keine Vorgaben macht, müssen bei der Unterstützung dieser Erweiterung zwingend Aspekte des Datenschutzes berücksichtigt und entsprechende Berechtigungskonzepte umgesetzt werden.

Außerdem befinden sich derzeit verschiedene Erweiterungen für OpenID in der Entwicklung. Zu ihnen gehören das OpenID Data Transport Protocol (DTP), das die Spezifikationen für OpenID Service Key Discovery [OpenID-SKD(D01)] und für OpenID DTP Messages [OpenID-DTPM(D03)] enthält, sowie Version 1.1 der OpenID Simple Registration Extension [OpenID-SRE(v1.1)] und ein OpenID Artifact Binding [OpenID-AB], das ein höheres Maß an Sicherheit verspricht. Seit Januar 2009 liegt auch ein Entwurf für eine Erweiterung namens OpenID OAuth Extension [OpenID-OAE] vor, die beschreibt wie man OpenID und OAuth [RFC5849] in Einklang bringt. In [Adid08] wurde vorgeschlagen, statt einer URL eine E-Mail-Adresse als Identifier zu verwenden, [CDS+08] enthält eine OpenID-Erweiterung für die Rechteverwaltung und Delegation und in [TaWa09] wurde schließlich gezeigt wie OpenID mit mobilen Endgeräten genutzt werden kann.

Sicherheit. Die Entwicklung von sicheren Browser-basierten Web-Applikationen und entsprechenden Single Sign-On Protokollen ist eine herausfordernde Aufgabe (vgl. [Slem01, Gros03, SAML-SecP(v2.0), GrPf06, GLS08, EHS09]). Deshalb ist es auch nicht verwunderlich, dass eine naive Realisierung des OpenID-Protokolls anfällig gegen Phishing, Man-in-the-Middle-Angriffe, Replay-Attacken, Cross-Site Request Forgery (CSRF) und Cross-Site Scripting (XSS) ist (vgl. [HySe08, Lind09, TsTs07]). Darüber hinaus wurde in [SKS10] ein Angriff gegen die Attribute Exchange Erweiterung [OpenID-AX(v1.0)] vorgestellt, die ausnutzt, dass die Anfragen beim OpenID-Protokoll nicht signiert werden.

Sofern der Benutzer jedoch ein starkes Authentisierungsverfahren verwendet, das TLS-Protokoll genutzt wird, die hierfür und für die Vereinbarung von $K_{RP,OP}$ und die Erstellung und Prüfung von $T = MAC(m, K_{RP,OP})$ genutzten Schlüssel authentisch sind und schließlich die Nachricht m, deren Integrität und Authentizität durch T geschützt wird, die sicherheitsrelevanten Daten[9] und alle Attribute umfasst, ist kein spezifischer Angriff gegen OpenID bekannt; ein formaler Sicherheitsbeweis im Stile von [GPS05, Gaje08, ACC+08] steht hierfür aber noch aus.

2.2 Elektronische Gesundheitskarte

Die elektronische Gesundheitskarte (eGK) ist eine in [eGK-1(v2.2.2), eGK-2(v2.2.1)] und [eGK-3(v2.2.0)] spezifizierte Chipkarte, die ein wesentliches Element der Sicherheitsar-

[9] Gemäß [OpenID-Auth(v2.0), Section 10.1] muss sich die „Signatur“ mindestens auf `op_endpoint`, `return_to`, `response_nonce`, `assoc_handle` sowie ggf. `claimed_id` und `identity` beziehen.

chitektur [FGHL07] der geplanten Telematikinfrastruktur für das deutsche Gesundheitswesen bildet. Technisch gesehen ist diese Chipkarte, deren Sicherheit durch eine Common Criteria Zertifizierung gemäß [BSI-PP-0020(v2.6)] nachgewiesen werden muss, insbesondere in der Lage die ausgefeilten in § 291a Abs. 4-5 [SGBV] definierten Zugriffsregeln für die darauf gespeicherten Patientendaten durchzusetzen und verschiedene kryptographische Operationen auszuführen.

Insbesondere enthält die eGK die folgenden Schlüssel, die grundsätzlich zur Authentisierung genutzt werden könnten:

- *PrK.eGK.AUT_CVC*
  Mit diesem privaten RSA-Schlüssel wird im Rahmen des in [eGK-1(v2.2.2), Abschnitt 16.2] spezifizierten, gegenseitigen Authentisierungsprotokolls, das typischer Weise mit einem Heilberufsausweis (HBA) [HBA-1(v2.3.2), HBA-2(v2.3.2)] oder einer Secure Module Card (SMC) [HBA-3(v2.3.2)] als Gegenstelle durchgeführt wird, die Echtheit der eGK nachgewiesen.

  Dieses zur gegenseitigen Authentisierung vorgesehene Protokoll besteht aus zwei weitgehend unabhängigen Teilprotokollen zur einseitigen Authentisierung im Stile des „Two-pass unilateral authentication protocol“ gemäß [ISO9798-3]. Da immer – also insbesondere auch ohne eine PIN-Eingabe und ohne die vorherige Authentisierung der Gegenseite – auf den privaten Schlüssel PrK.eGK.AUT_CVC zugegriffen werden kann (vgl. [eGK-2(v2.2.1), Abschnitt 6.2.9]), kann mit diesem Schlüssel ein besonders benutzerfreundliches, einseitiges Authentisierungsprotokoll realisiert werden, bei dem der Benutzer nur die eGK einstecken aber keine PIN eingeben muss.

  Dieses Authentisierungsprotokoll besteht aus folgenden Schritten:

  1. Erzeugen einer Zufallszahl r.
  2. Bilden eines APDU-Stapels, durch den
     - die CV-Zertifikate C.CA_eGK.CS und C.eGK.AUT_CVC von der eGK gelesen werden und
     - die Zufallszahl r mit dem privaten Schlüssel PrK.eGK.AUT_CVC signiert wird.
  3. Übermitteln des APDU-Stapels zur eGK mit der `Transmit`-Funktion aus [BSI-TR-03112(v1.1)], wodurch man in `TransmitResponse` die CV-Zertifikate und eine gemäß [ISO9796-2] DS1 gebildete Signatur s erhält.
  4. Prüfen der CV-Zertifikate gegen den Wurzelschlüssel der gematik.
  5. Prüfen der Signatur s über die Zufallszahl r.

- *PrK.CH.AUT und PrK.CH.AUTN*
  Die beiden privaten RSA-Schlüssel PrK.CH.AUT (vgl. [eGK-2(v2.2.1), Abschnitt 6.4.6]) und PrK.CH.AUTN (vgl. [eGK-2(v2.2.1), Abschnitt 6.4.7]) können beispielsweise nach Eingabe der PIN.home für die Authentisierung gemäß [ISO9798-3] genutzt werden. Da für diese beiden Schlüssel auch entsprechende X.509-Zertifikate

(vgl. [eGK-2(v2.2.1), Abschnitte 6.4.1-6.4.2] und [gemX.509-eGK(v1.5.9)]) zur Verfügung stehen, könnte damit auch eine TLS-Client-Authentisierung gemäß [RFC5246] durchgeführt werden. Außerdem kann mit dem Online Certificate Status Protocol (OCSP) gemäß [RFC2560] der Sperrstatus dieser Zertifikate bzw. der eGK ermittelt werden.

- *PrK.CH.ENC und PrK.CH.ENCV*
  Die beiden privaten RSA-Schlüssel PrK.CH.ENC (vgl. [eGK-2(v2.2.1), Abschnitt 6.4.8]) und PrK.CH.ENCV (vgl. [eGK-2(v2.2.1), Abschnitt 6.4.9]) können beispielsweise nach Eingabe der PIN.home zur Entschlüsselung von Daten genutzt werden, wodurch ein Authentisierungsprotokoll gemäß [NeSc78, Lowe96] realisiert werden könnte.

Außerdem sind auf der eGK weitere geheime Schlüssel (SK.CMS,SK.VSD und SK.VSDCMS, siehe [eGK-2(v2.2.1), Abschnitt 6.2.11-6.2.13]) vorhanden, die jedoch nur zur Authentisierung gegenüber dem Kartenmanagementsystem bzw. dem Versichertenstammdatendienst der Krankenkasse genutzt werden können.

Schließlich kann auf der eGK ein privater Signaturschlüssel Prk.CH.QES (vgl. [eGK-2(v2.2.1), Abschnitt 7.1.3 und 7.7.7]) vorhanden sein, der aber aus nahe liegenden Gründen nicht zur Authentisierung genutzt werden sollte.

3 OpeneGK

3.1 Überblick

Bei der in diesem Papier vorgeschlagenen Kombination der elektronischen Gesundheitskarte mit dem OpenID-Protokoll ist der Bürger, wie in Abbildung 2 dargestellt, mit einer elektronischen Gesundheitskarte (siehe Abschnitt 2.2), einem entsprechenden Kartenterminal und einem Bürgerclient (siehe Abschnitt 3.2) ausgestattet. Um sich bei einem Mehrwertdienst (siehe Abschnitt 3.3) zu registrieren und authentisieren läuft das in Abschnitt 2.1 beschriebene Protokoll ab, bei dem der OpeneGK-Dienst (siehe Abschnitt 3.4) in Schritt (5) auf die elektronische Gesundheitskarte zugreift.

3.2 Bürgerclient

Der Bürgerclient ist eine Chipkarten-Middleware, die derzeit im Auftrag des Bundesinnenministeriums entwickelt wird [Heise091116] und alle Chipkarten der eCard-Strategie [eCard-PM, Kowa07] – also insbesondere auch die elektronische Gesundheitskarte – unterstützt. Der Bürgerclient setzt das in [BSI-TR-03112(v1.1)] spezifizierte eCard-API-Framework um, das wiederum auf einer Vielzahl von internationalen Standards basiert [HuBa08].

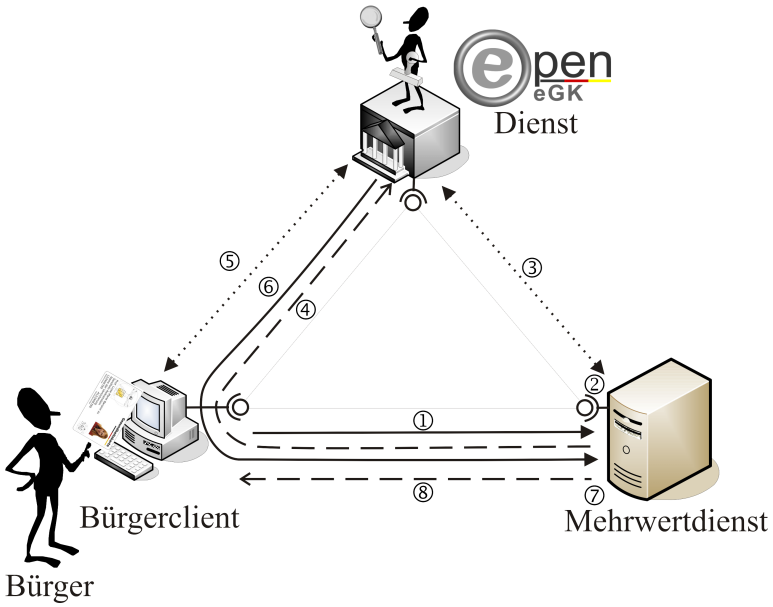

Abbildung 2: Authentisierung und Registrierung mit OpeneGK

3.3 Mehrwertdienst

Der kritische Faktor für die schnelle Verbreitung (vgl. [HRZ10]) eines Protokolls wie OpenID [OpenID-Auth(v2.0)] ist vor allem dessen Einfachheit verglichen mit den konkurrierenden Protokollen wie beispielsweise SAML [SAML(v2.0)]. Dabei geht es nicht nur um den Protokollablauf selbst, sondern vor allem auch um die breite Verfügbarkeit und leichte Integrierbarkeit der erforderlichen Programmbibliotheken. Da für alle relevanten Programmiersprachen bereits entsprechende OpenID-Bibliotheken existieren[10], könnte der OpeneGK-Dienst beispielsweise unter Verwendung einer dieser Bibliotheken in einen Mehrwertdienst integriert werden. Allerdings operieren diese Bibliotheken zumeist auf einem vergleichsweise niedrigen Abstraktionsniveau und der Entwickler des Mehrwertdienstes bzw. ein Integrator müsste sowohl die genauen Protokollabläufe als auch die damit verbundenen Sicherheitsaspekte (vgl. [HySe08, Lind09, TsTs07]) kennen, um eine zuverlässige Benutzerauthentifizierung sicher zu stellen.

Um die sichere Anbindung des OpeneGK-Dienstes zu erleichtern soll im Folgenden ein Schnittstellenentwurf für eine erweiterte Bibliothek vorgestellt werden, die mit Version 1.1 und 2.0 der OpenID Spezifikation [OpenID-Auth(v1.1), OpenID-Auth(v2.0)] und deren Erweiterungen [OpenID-SRE(v1.0), OpenID-PAPE(v1.0), OpenID-AX(v1.0)] kom-

[10] Siehe `http://openid.net/developers/libraries`.

patibel ist, eine besonders einfache Integration ermöglicht, sinnvolle Voreinstellungen für die sichere Authentisierung mit der eGK mitbringt und schließlich die verschiedenen Policies und speziellen Features des OpeneGK-Dienstes, wie z.B. die Erzeugung von Mehrwertdienst-spezifischen Pseudonymen (vgl. Abschnitt 3.4), unterstützt.

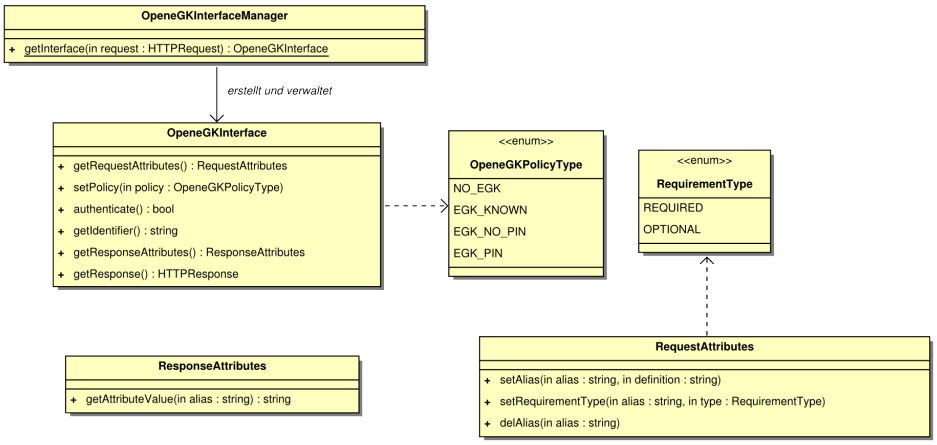

Abbildung 3: UML-Design der OpeneGK-Schnittstelle

Abbildung 3 zeigt ein UML Klassendiagramm mit allen Klassen der OpeneGK-Schnittstelle. Die gesamte Protokolllogik wird von `OpeneGKInterface` zur Verfügung gestellt. Wie in Abschnitt 2.1 erläutert, sind für eine Registrierung oder Authentisierung beim OpenID-Protokoll zwei HTTP Nachrichten zwischen dem Mehrwertdienst (Relying Party) und dem Bürgerclient (User Agent) nötig. Um die HTTP-Sitzungsverwaltung zu erleichtern kümmert sich die Klasse `OpeneGKInterfaceManager` um die Erstellung und Zuordnung der `OpeneGKInterface`-Instanzen zu den HTTP-Sitzungen.

Während des Authentisierungsprozesses stellt die OpeneGK-Schnittstelle Möglichkeiten bereit, um spezifische Einstellungen vorzunehmen und Ergebnisdaten abzufragen. Die Klasse `RequestAttributes` bietet beispielsweise die Möglichkeit die in einer Konfigurationsdatei definierten Standardwerte für die gewünschten Attribute [11] zu überschreiben. Mit der Funktion `setAlias` werden den Attributdefinitionen kurze Alias-Namen und URLs zugewiesen, um nach der Authentisierung bequem auf diese zugreifen zu können. Das OpenID-Protokoll sieht für angefragte Attribute vor, dass diese nicht zwingend in der Antwort des OpenID-Providers enthalten sein müssen. Mit den Werten aus `Requirement-Type` können bestimmte Attribute als `OPTIONAL` markiert werden, wodurch der Standardwert (`REQUIRED`) überschrieben wird.

Die gewünschte Authentisierungsmethode wird mit den vordefinierten Werten aus `OpeneGKPolicyType` gesteuert, die eine eGK-spezifische Authentication Policy im Sinne von [OpenID-PAPE(v1.0)] umsetzen. Ebenso wie bei den Attributen wird die benötigte Policy im Regelfall nur durch die Konfigurationsdatei bestimmt.

[11] Siehe [OpenID-SRE(v1.0)] und [OpenID-AX(v1.0)]

Nach einer erfolgreichen Authentisierung wird durch die Klasse `ResponseAttributes` eine komfortable Schnittstelle zum Ausgeben der Attribute bereitgestellt. Zu einem Alias wird entweder der empfangene Wert, oder ein sprachspezifisches leeres Symbol zurückgeliefert.

Nachdem die Möglichkeiten der Einflussnahme der Applikation auf den Authentisierungsprozess beschrieben wurde, bleibt es noch den Ablauf unter Verwendung der Schnittstelle zu beschreiben. In Anlehnung an die Schritte aus Abbildung 1 in Abschnitt 2.1 ergibt sich folgender Ablauf:

- Schritt (1)-(4):

  Der Prozess wird dadurch gestartet, dass der Mehrwertdienst eine Authentisierungsanfrage in Form einer HTTP GET Nachricht empfängt. An die HTTP Nachricht werden zwei Anforderungen gestellt. Zum Einen muss der Parameter `return_to`, der einen Verweis zu der ursprünglich angefragten Ressource darstellt, gesetzt sein. Zum Anderen muss in der Nachricht eine HTTP Sitzung zu finden sein anhand derer die richtige `OpeneGKInterface`-Instanz ausgewählt werden kann. Anders als bei einer gewöhnlichen OpenID-Authentisierung ist der Wert `openid.identity` optional. Fehlt er, so wird automatisch eine eGK-basierte Authentisierung angenommen und die Identität des Nutzers unter Verwendung der eGK ermittelt.

  In diesem Schritt sind folgende Funktionsaufrufe zu tätigen.

  a) `OpeneGKInterfaceManager.getInterface`
  Durch diesen Aufruf wird mit Informationen aus der HTTP Nachricht eine Instanz der Schnittstelle erstellt. Im Nachfolgenden werden Funktionen aus der Klasse `OpeneGKInterface` immer mit dieser Instanz ausgeführt.

  b) `getRequestAttributes` und `setPolicy` (optional)
  Nun können optional die konfigurierten Standardwerte für die anzufordernden Attribute (mit `getRequestAttributes`) oder die gewünschte Policy (mit `setPolicy`) (vgl. Abschnitt 3.4) überschrieben werden.

  c) `getHTTPResponse`
  Sofern noch keine Sicherheitsbeziehung zwischen dem Mehrwertdienst und dem OpeneGK-Dienst vorhanden ist, wird diese etabliert und danach die HTTP Antwort mit der OpenID-Anfrage erzeugt und zurückgegeben. Die Funktion stellt auch das vorläufige Ende des Prozesses dar, da nun der OpeneGK-Dienst für die Authentisierung in Schritt (5) zuständig ist und der Mehrwertdienst auf eine neue HTTP Anfrage wartet.

- Schritt (6)-(8):

  Die folgenden Schritte laufen ab, sobald der Mehrwertdienst eine HTTP GET Anfrage mit einer OpenID-Authentisierungsantwort empfängt:

  e) `OpeneGKInterfaceManager.getInterface`
  Statt wie im vorherigen Schritt eine neue Instanz zu erzeugen, wird hier die der HTTP-Sitzung zugehörige Instanz zurückgeliefert.

f) `authenticate`
In diesem Schritt wird die OpenID-Antwort ausgewertet, das Authentisierungstoken T geprüft und das Ergebnis der Authentifizierung zurückgeliefert.

g) `getIdentifier` (optional)
Im Fall einer erfolgreichen Authentifizierung kann, sofern nicht eine völlig anonyme Nutzung des Mehrwertdienstes vorgesehen ist, mit dieser Funktion das Dienst-spezifische Pseudonym des Benutzers zurückgeliefert werden.

h) `getResponseAttributes` (optional)
Aus der bereits oben beschriebenen Struktur können ggf. die zur Registrierung in der Applikation notwendigen Attribute extrahiert werden.

i) `getHTTPResponse`
Den Abschluss der OpenID-Kommunikation markiert die an den Bürgerclient geschickte HTTP-Antwort, wodurch eine Umleitung des Bürgerclients auf die in der ersten Anfrage gesendete `return_to`-URL erfolgt.

3.4 OpeneGK-Dienst

Der OpeneGK-Dienst ist einerseits ein OpenID-Provider, der über das in [OpenID-Auth(v2.0)] definierte Protokoll angesprochen werden kann. Andererseits umfasst der OpeneGK-Dienst ein „serverseitiges eCard-API-Framework“ [BSI-TR-03112(v1.1)] über das mit dem Bürgerclient kommuniziert und letztlich auf die elektronische Gesundheitskarte zugegriffen werden kann.

Der OpeneGK-Dienst unterstützt die folgenden Authentication Policies (vgl. Abbildung 3):

- `NO_EGK`
  In diesem Fall, der in Verbindung mit einem beliebigen OpenID-Provider genutzt werden kann, wurde die elektronische Gesundheitskarte weder zur erstmaligen Registrierung noch zur aktuellen Authentisierung genutzt. Durch diese Policy können Mehrwertdienste die OpeneGK-Schnittstelle und ggf. den OpeneGK-Dienst bereits nutzen obwohl die elektronische Gesundheitskarte noch gar nicht flächendeckend ausgerollt ist.

- `EGK_KNOWN`
  In diesem Fall wurde zwar die erstmalige Registrierung mit der elektronischen Gesundheitskarte durchgeführt aber die aktuelle Authentisierung erfolgte am OpeneGK-Dienst unter Verwendung eines alternativen Authentisierungsverfahrens. Dadurch kann der OpeneGK-Dienst zur Authentisierung auch dann genutzt werden, wenn gerade kein kontaktbehaftetes Chipkartenterminal für die direkte Nutzung der eGK zur Hand ist.

- `EGK_NO_PIN`
  Bei dieser Policy erfolgt die Authentisierung mit dem für die Card-2-Card-Authenti-

sierung vorgesehenen Schlüssel PrK.eGK.AUT_CVC. Wie in [eGK-2(v2.2.1), Abschnitt 6.2.9] spezifiziert, ist für die Nutzung dieses Schlüssels keine PIN-Eingabe erforderlich, so dass die eGK zur Authentisierung einfach gesteckt sein muss und eine besonders komfortable Authentisierung möglich wird.

- `EGK_PIN`
  Bei dieser Authentisierungsvariante wird nach Eingabe der PIN der private Schlüssel PrK.CH.AUTN (vgl. [eGK-2(v2.2.1), Abschnitt 6.4.7]) zur Authentisierung genutzt. Da bei dieser Variante auch der Sperrstatus der eGK geprüft werden kann, bietet diese Authentication Policy das größte Maß an Sicherheit.

Ein weiterer Vorteil des OpeneGK-Dienstes im Vergleich zur naiven Nutzung von OpenID oder der elektronischen Gesundheitskarte liegt in der Möglichkeit Mehrwertdienstspezifische Pseudonyme zu konstruieren, die einen Benutzer am Mehrwertdienst selbst nach einem Krankenkassen- und Kartenwechsel hinweg eindeutig wiedererkennen lassen aber eine Dienst-übergreifende Verkettung der Pseudonyme und dadurch die Profilbildung unmöglich machen. Dies kann beim OpeneGK-Dienst beispielsweise dadurch erreicht werden, dass der im C.CH.AUT-Zertifikat (vgl. [eGK-2(v2.2.1), Abschnitt 6.4.1] und [gemX.509-eGK(v1.5.0), Abschnitt 6]) enthaltene unveränderbare Teil der Krankenversicherungsnummer des Versicherten (vgl. [gemX.509-eGK(v1.5.0), Abschnitt 5.6]), die Domain des Mehrwertdienstes und ein OpeneGK-spezifisches Geheimnis konkateniert und daraus ein kryptographischer Hashwert gebildet wird. Eine standardmäßige Verwendung von solchen Pseudonymen wäre aufgrund der Einfachheit für alle Mehrwertdienste zu empfehlen.

4 Diskussion

Der OpenID-Ansatz ist zunächst primär für den Benutzer vorteilhaft: er muss sich nicht wieder neue Zugangsdaten merken und kann mit einem Konto sich bei verschiedenen Websites anmelden. Diese Idee ist nicht besonders neu (vgl. [Berr98, HiWi00]), wurde aber bei OpenID besonders neutral, flexibel und dabei für alle Seiten besonders einfach umgesetzt. Für den Service-Provider ergibt sich damit indessen lediglich ein Marketing-Effekt. Er wird für die Nutzer etwas attraktiver und kann so eventuell seine Kundenzahl erhöhen. Technisch bedeutet es für ihn die Auslagerung des Benutzermanagements, was eine gewisse Vereinfachung der Verwaltung darstellt.

Unser Vorschlag einer OpeneGK-Anmeldung bringt dem Benutzer bereits ähnliche Vorteile wie bei OpenID allein, beinhaltet jedoch für den Dienstanbieter einen echten Mehrwert. Er weiß dadurch, dass es sich bei den sich damit anmeldenden Bürgern um Personen handelt, deren Identität und persönliche Daten zuverlässig verifiziert wurden. Verschiedene Missbrauchsszenarien sind damit von vornherein verhindert; der Service-Provider kann solchen Nutzern sogar durch das Anbieten weitergehender Dienste mehr Vertrauen entgegen bringen.

Der Bürger hat aber gleichfalls zusätzliche Vorteile. Er ist beispielsweise nicht gezwungen

sich mit ein und demselben Benutzernamen bei allen angeschlossenen Sites anzumelden. Der OpeneGK-Ansatz erlaubt es, für jeden Service-Provider individuelle Pseudonyme zu verwenden, die auf Wunsch sogar automatisch generiert werden können. So bleibt die Privatsspähre des Benutzers auch dann gewahrt, wenn er den gleichen Authentisierungsmechanismus bei mehreren untereinander verbundenen Anbietern nutzt.

Angesichts der aktuellen politischen Diskussionen um die elektronische Gesundheitskarte steckt im OpeneGK-Vorschlag aber noch ein ganz pragmatischer Vorteil: Diese Technik setzt nur auf die Karte selbst auf und kann daher bereits starten und genutzt werden, bevor die geplante Infrastruktur der gematik verfügbar ist. Denn wann Letzteres soweit ist, ist Stand heute nicht abzuschätzen.

Bei der Konzeption von OpeneGK wurde besonderer Wert auf sichere Kommunikation und Datenschutz gelegt. Um dies an allen Stellen zu gewährleisten, ist natürlich eine korrekte Anbindung dieses Dienstes bei einem Service-Provider zu realisieren. Insbesondere ist ein Schlüsselaustausch zwischen OpeneGK-Provider und Service-Provider zwingend erforderlich.

Es sind nur wenige sensible medizinische Daten auf der eGK vorgesehen, überdies nur freiwillig. Für den OpeneGK-Dienst werden diese aber nicht einmal benötigt und daher auch nicht verwendet. Er sorgt nur für die Registrierung, Authentisierung und Identifikation; medizinische Angaben muss der Benutzer sofern gewünscht direkt mit dem Service-Provider austauschen. Daher ist die Hemmschwelle für die Anwender sicher als gering einzustufen, da sie kein Datenschutzrisiko eingehen.

5 Fazit

Gerade im medizinischen Bereich geht es oft um individuelle Daten, bei denen jeder von uns selbst darüber bestimmen will, welche er an wen weiter gibt. Durch die starke Authentisierung bei OpeneGK wird sicher gestellt, dass Unbefugte nicht einfach durch Ausspähen eines Passworts an diese Daten gelangen können. Für die Dienstanbieter heißt das, dass ihnen schrittweise mehr Vertrauen geschenkt werden wird, so dass die Kunden eher bereit sind, individuelle medizinische Daten bei ihnen zu hinterlegen. Der Datenschutz kann so signifikant erhöht werden. Und das alles kann mit einer Karte, die in absehbarer Zeit ein Großteil der bundesdeutschen Bevölkerung ohnehin bei sich tragen wird, umgesetzt werden.

Das Konzept, das OpenID-Protokoll um die Verwendung einer Chipkarte zur starken Authentisierung zu erweitern, ist im medizinischen Umfeld besonders attraktiv, aber keineswegs darauf beschränkt. Neben der eGK könnte auch eine andere staatliche Identitätskarte verwendet werden, wie sie in vielen europäischen Ländern geplant ist und bereits verwendet wird. In Deutschland ist dies der neue Personalausweis, der ab Ende 2010 ausgegeben wird. Diese Option macht den oben beschriebenen Ansatz noch flexibler und ermöglicht eine Vielzahl weiterer Anwendungsfälle.

Auf der anderen Seite hat OpeneGK den Vorteil, dass die Schnittstelle bewusst einfach und generisch gehalten ist. Dies erlaubt es prinzipiell, das Konzept auch für andere Fra-

meworks für Single-Sign-On wie SAML oder CardSpace [BSB08] nutzbar zu machen. Dies würde technisch zwar eine deutlich komplexere Realisierung bedeuten, kann für die Service-Provider aber weitgehend transparent bleiben.

Somit stellt OpeneGK einen sehr flexiblen Ansatz dar, der großes Potenzial zur Erweiterung in vielerlei Hinsicht in sich trägt.

Literatur

[ACC+08] A. ARMANDO, R. CARBONE, L. COMPAGNA, J. CUELLAR, und L. TOBARRA. *Formal analysis of SAML 2.0 web browser single sign-on: breaking the SAML-based single sign-on for google apps*. ACM Workshop on Formal Methods in Security Engineering. http://www.ai-lab.it/armando/pub/fmse9-armando.pdf, 2008.

[Adid08] BEN ADIDAD. *EmID: Web Authentication by Email Address*. http://ben.adida.net/research/w2sp2008-emid.pdf, 2008.

[Berr98] PHILIPPE LE BERRE. *Authentication between servers*. European Patent Application EP 0 940 960 A1, March 1998.

[BSB08] V. BERTOCCI, G. SERACK, und C. BAKER. *Understanding Windows CardSpace - An Introduction to the Concepts and Challenges of Digital Identities* (Addison Wesley, 2008).

[BSI-PP-0020(v2.6)] BUNDESAMT FÜR SICHERHEIT IN DER INFORMATIONSTECHNIK. *Common Criteria Protection Profile–electronic Health Card (eHC)–elektronische Gesundheitskarte (eGK)*. BSI-PP-0020-V2-2007-MA02, Version 2.6, 29.07.2008. http://www.gematik.de/upload/gematik_eGK_Specification_Part2_e_V1%_1_1_516.pdf, 2008.

[BSI-TR-03112(v1.1)] FEDERAL OFFICE FOR INFORMATION SECURITY (BUNDESAMT FÜR SICHERHEIT IN DER INFORMATIONSTECHNIK). *eCard-API-Framework*. Technical Directive (BSI-TR-03112), Version 1.1, Part 1-7. https://www.bsi.bund.de/ContentBSI/Publikationen/TechnischeRichtlinien/tr03112/index_htm.html, 2009.

[CDS+08] BRYANT CUTLER, DEVLIN DALEY, KENT SEAMONS, und PHIL WINDLEY. *SimplePermissions: an OpenID Extension for Delegation and Permissions Model Discovery*. http://www.eclab.byu.edu/simplepermissions_techreport.pdf, 2008.

[eCard-PM] BUNDESREGIERUNG. *eCard-Strategie der Bundesregierung*. Pressemitteilung vom 09.03.2005. http://www.bmwi.de/Navigation/Presse/pressemitteilungen,did=60006.html, 2005.

[eGK-1(v2.2.2)] GESELLSCHAFT FÜR TELEMATIKANWENDUNGEN DER GESUNDHEITSKARTE (GEMATIK). *Die Spezifikation der elektronischen Gesundheitskarte - Teil 1: Spezifikation der elektrischen Schnittstelle*. Version 2.2.2 vom 16.09.2008. http://www.gematik.de/upload/gematik_eGK_Spezifikation_Teil1_V2_2_2_4411.pdf, 2008.

[eGK-2(v2.2.1)] GESELLSCHAFT FÜR TELEMATIKANWENDUNGEN DER GESUNDHEITSKARTE (GEMATIK). *Die Spezifikation der elektronischen Gesundheitskarte - Teil 2: Grundlegende Applikationen*. Version 2.2.1 vom 16.09.2008. http://www.gematik.de/upload/gematik_eGK_Spezifikation_Teil2_V2_2_1_3805.pdf, 2008.

[eGK-3(v2.2.0)] GESELLSCHAFT FÜR TELEMATIKANWENDUNGEN DER GESUNDHEITSKARTE (GEMATIK). *Die Spezifikation der elektronischen Gesundheitskarte - Teil 3: Äußere Gestaltung*. Version 2.2.0 vom 02.07.2008. http://www.gematik.de/upload/gematik_eGK_Spezifikation_Teil3_V2_2_0_3806.pdf, 2008.

[EHS09] JAN EICHHOLZ, DETLEF HÜHNLEIN, und JÖRG SCHWENK. *SAMLizing the European Citizen Card.* In *Proceedings of* BIOSIG 2009: Biometrics and Electronic Signatures, Band 155 von *Lecture Notes in Informatics (LNI)*, Seiten 105–117 (GI-Edition, 2009). http://www.ecsec.de/pub/SAMLizing-ECC.pdf.

[EHS10] JAN EICHHOLZ AND DETLEF HÜHNLEIN AND JOHANNES SCHMÖLZ. *A SAML-profile for electronic identity cards*. to appear, 2010.

[FGHL07] FLORIAN FANKHAUSER, THOMAS GRECHENIG, DETLEF HÜHNLEIN, und MANFRED LOHMAIER. *Die Basiskonzepte der Sicherheitsarchitektur bei der Einführung der eGK*. In PATRICK HORSTER (Herausgeber), *Tagungsband DACH Security 2007*, Seiten 326–337 (IT-Verlag, 2007). http://www.ecsec.de/pub/2007_DACH_eGK-Sicherheitsarchitektur.pdf.

[gemX.509-eGK(v1.5.0)] GESELLSCHAFT FÜR TELEMATIKANWENDUNGEN DER GESUNDHEITSKARTE (GEMATIK). *Festlegungen zu den X.509 Zertifikaten der Versicherten*. Version 1.5.0 vom 12.06.2008. http://www.gematik.de/upload/gematik_PKI_X509_Zertifikate_des_Versicherten_eGK_V1.5.0_3854.pdf, 2008.

[gemX.509-eGK(v1.5.9)] GESELLSCHAFT FÜR TELEMATIKANWENDUNGEN DER GESUNDHEITSKARTE (GEMATIK). *Festlegungen zu den X.509 Zertifikaten der Versicherten*. Version 1.5.9 vom 03.07.2009, 2009.

[Gaje08] SEBASTIAN GAJEK. *A Universally Composable Framework for the Analysis of Browser-Based Security Protocols*. In JOONSANG BAEK, FENG BAO, KEFEI CHEN, und XUEJIA LAI (Herausgeber), *Provable Security – Second International Conference, ProvSec 2008, Shanghai, China, October 30 - November 1*, Band 5324 von *Lecture Notes in Computer Science*, Seiten 283–297 (Springer, 2008).

[GLS08] JÖRG SCHWENK, LIJUN LIAO, und SEBASTAN GAJEK. *Stronger Bindings for SAML Assertions and SAML Artifacts*. In *Proceedings of the 5th ACM CCS Workshop on Secure Web Services (SWS'08)*, Seiten 11–20 (ACM Press, 2008).

[GPS05] THOMAS GROSS, BIRGIT PFITZMANN, und AHMAD-REZA SADEGHI. *Browser Model for Security Analysis of Browser-Based Protocols*. In *ESORICS: 10th European Symposium on Research in Computer Security*, Band 3679, Seiten 489–508 (Berlin, Germany, 2005). http://eprint.iacr.org/2005/127.pdf.

[Gros03] THOMAS GROSS. *Security Analysis of the SAML Single Sign-on Browser/Artifact Profile*. In *Annual Computer Security Applications Conference, December 8-12, 2003, Aladdin Resort & Casino Las Vegas, Nevada, USA* (2003). http://www.acsac.org/2003/papers/73.pdf.

[GrPf06] THOMAS GROSS und BIRGIT PFITZMANN. *SAML Artifact Information Flow Revisited*. In *In IEEE Workshop on Web Services Security (WSSS)*, Seiten 84–100 (IEEE, Berkeley, 2006). http://www.zurich.ibm.com/security/publications/2006/GrPf06.SAML-Artifacts.rz3643.pdf.

[HBA-1(v2.3.2)] BUNDESÄRZTEKAMMER ET. AL. *German Health Professional Card and Security Module Card – Part 1: Commands, Algorithms and Functions of the COS Platform*. Version 2.3.2, 05.08.2009. http://www.bundesaerztekammer.de/downloads/HPC-Spezifikation_2.3.2_-_COS_Teil_1_.pdf, 2009.

[HBA-2(v2.3.2)] BUNDESÄRZTEKAMMER ET. AL. *German Health Professional Card and Security Module Card – Part 2: HPC Applications and Functions*. Version 2.3.2, 05.08.2009. http://www.bundesaerztekammer.de/downloads/HPC-Spezifikation_2.3.2_-_HPC_Teil_2_.pdf, 2009.

[HBA-3(v2.3.2)] BUNDESÄRZTEKAMMER ET. AL. *German Health Professional Card and Security Module Card – Part 3: SMC Applications and Functions*. Version 2.3.2, 05.08.2009. http://www.bundesaerztekammer.de/downloads/HPC-Spezifikation_2.3.2_-_SMC_Teil_3_.pdf, 2009.

[Heise091116] HEISE. *Elektronischer Personalausweis: Bürger-Client auf dem Weg zum Nutzer*. Meldung vom 16.11.2009, 15:13 Uhr. http://tinyurl.com/yz4kzno, 2009.

[HiWi00] HEATHER MARIA HINTON und DAVID JOHN WINTERS. *Method and system for web-based cross-domain single-sign-on authentication*. World-wide patent, WO 02/39237 A2, November 2000.

[HRZ10] DETLEF HÜHNLEIN, HEIKO ROSSNAGEL, und JAN ZIBUSCHKA. *Diffusion of Federated Identity Management*. to appear, 2010.

[HuBa08] DETLEF HÜHNLEIN und MANUEL BACH. *Die Standards des eCard-API-Frameworks – Eine deutsche Richtlinie im Konzert internationaler Normen. Datenschutz und Datensicherheit (DuD)*, (6):379–384. http://www.ecsec.de/pub/2008_DuD_eCard.pdf, 2008.

[HySe08] HYUN-KYUNG-OH und SEUNG-HUN-JIN. *The security limitations of SSO in OpenID*. In *2008 10th International Conference on Advanced Communication Technology, Gangwon-Do, South Korea, 17-20 Feb. 2008*, Seiten 1608–1611 (IEEE, 2008). http://mnet.skku.ac.kr/data/2008data/ICACT2008/pdf/tech/08F-03.pdf.

[ID-MI(v1.0)] MICHAEL B. JONES und MICHAEL MCINTOSH. *Identity Metasystem Interoperability Version 1.0*. OASIS Standard. http://docs.oasis-open.org/imi/identity/v1.0/os/identity-1.0-spec-os.pdf, July 2009.

[ISO9796-2] *ISO-IEC 9796-2: Information Technology - Security Techniques - Digital Signature Schemes Giving Message Recovery - Part 2: Integer Factorization Based Mechanisms*. International Standard, Oktober 2002.

[ISO9798-3] *ISO-IEC 9798-3: Information Technology – Security Techniques – Entity Authentication – Part 3: Mechanisms using digital signature techniques*. International Standard, 1998.

[Kiss09] BRIAN KISSEL. *OpenID 2009 Year in Review*. December 16, 2009. http://openid.net/2009/12/16/openid-2009-year-in-review/.

[Kowa07] BERND KOWALSKI. *Die eCard-Strategie der Bundesregierung im Überblick*. In D. HÜHNLEIN A. BRÖMME, C. BUSCH (Herausgeber), *BIOSIG 2007: Biometrics and Electronic Signatures, Proceedings of the Special Interest Group on Biometrics and Electronic Signatures*, Band 108 von *LNI*, Seiten 87–96 (2007).

[Lind09] ALEXANDER LINDHOLM. *Security Evaluation of the OpenID Protocol*. Master-Thesis, Royal Institute of Technology, School of Computer Science and Communication, KTH CSC, Stockholm. http://w3.nada.kth.se/utbildning/grukth/exjobb/rapportlistor/2009/rapporter09/lindholm_alexander_09076.pdf, 2009.

[Lowe96] GAVIN LOWE. *Breaking and Fixing the Needham-Schroeder Public-Key Protocol Using FDR*. In TIZIANA MARGARIA und BERNHARD STEFFEN (Herausgeber), *Tools and Algorithms for Construction and Analysis of Systems, Second International Workshop, TACAS '96, Passau, Germany, March 27-29, 1996, Proceedings*, Band 1055 von *Lecture Notes in Computer Science*, Seiten 147–166 (Springer, 1996).

[NeSc78] ROGER NEEDHAM und MICHAEL D. SCHROEDER. *Using encryption for authentication in large networks of computers*. *Communications of the ACM*, Band 21(12), 1978.

[NIST-800-63] NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY. *Electronic Authentication Guideline*. NIST Special Publication 800-63 Version 1.0.2. http://csrc.nist.gov/publications/nistpubs/800-63/SP800-63V1_0_2.pdf.

[OpenID-AB] N. SAKIMURA J. BRADLEY. *OpenID Artifact Binding 1.0*. Draft07, 14.05.2010. http://www.sakimura.org/specs/ab/1.0/.

[OpenID-AX(v1.0)] OPENID FOUNDATION. *OpenID Attribute Exchange 1.0*. Final, December 5, 2007. http://openid.net/specs/openid-attribute-exchange-1_0.html.

[OpenID-Auth(v1.1)] D. RECORDON und B. FITZPATRICK. *OpenID Authentication 1.1*. May 2006. http://openid.net/specs/openid-authentication-1_1.html.

[OpenID-Auth(v2.0)] OPENID FOUNDATION. *OpenID Authentication 2.0*. Final, December 5, 2007. http://openid.net/specs/openid-authentication-2_0.html.

[OpenID-DTPM(D03)] OPENID FOUNDATION. *OpenID DTP Messages 1.0 - Draft 03*. Draft, December 06, 2006. http://openid.net/specs/openid-dtp-messages-1_0-03.html.

[OpenID-OAE] D. BALFANZ, B. DE MEDEIROS, D. RECORDON, J. SMARR, und A. TOM. *OpenID OAuth Extension*. Draft, January 7, 2009. http://step2.googlecode.com/svn/spec/openid_oauth_extension/latest/openid_oauth_extension.html, 2009.

[OpenID-PAPE(v1.0)] OPENID FOUNDATION. *OpenID Provider Authentication Policy Extension 1.0*. December 30, 2008. http://openid.net/specs/openid-provider-authentication-policy-extension-1_0.html.

[OpenID-Pro] OPENID FOUNDATION. *Get an OpenID*. http://openid.net/get-an-openid, 2010.

[OpenID-SKD(D01)] OPENID FOUNDATION. *OpenID Service Key Discovery 1.0 - Draft 01*. Draft, December 06, 2006. http://openid.net/specs/openid-service-key-discovery-1_0-01.html.

[OpenID-SRE(v1.0)] OPENID FOUNDATION. *OpenID Simple Registration Extension 1.0*. June 30, 2006. http://openid.net/specs/openid-simple-registration-extension-1_0.html.

[OpenID-SRE(v1.1)] OPENID FOUNDATION. *OpenID Simple Registration Extension 1.1 - Draft 1*. Draft, December 06, 2006. http://openid.net/specs/openid-simple-registration-extension-1_1-01.html.

[Raep09] MARTIN RAEPPLE. *Netzweite Identitäten mit OpenID*. *Datenschutz und Datensicherheit (DuD)*, Band 33(3):174–177. http://www.springerlink.com/content/755180g7h7741187/, 2009.

[RCT08] DRUMMOND REED, LES CHASEN, und WILLIAM TAN. *OpenID identity discovery with XRI and XRDS*. In *IDtrust '08: Proceedings of the 7th symposium on Identity and trust on the Internet*, Seiten 19–25 (ACM, 2008). http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.123.4747&rep=rep1&type=pdf.

[RFC2560] M. MYERS, R. ANKNEY, A. MALPANI, S. GALPERIN, und C. ADAMS. *X.509 Internet Public Key Infrastructure – Online Certificate Status Protocol - OCSP*. Request For Comments – RFC 2560. http://www.ietf.org/rfc/rfc2560.txt, 1999.

[RFC5246] T. DIERKS und E. RESCORLA. *The Transport Layer Security (TLS) Protocol Version 1.2*. Request For Comments – RFC 5246. http://www.ietf.org/rfc/rfc5246.txt, August 2008.

[RFC5849] E. HAMMER-LAHAV. *The OAuth 1.0 Protocol*. Request For Comments – RFC 5849. http://www.ietf.org/rfc/rfc5849.txt, April 2010.

[Saki10] NAT SAKIMURA. *NTT docomo is now an OpenID Provider*. March 9, 2010. http://openid.net/2010/03/09/ntt-docomo-is-now-an-openid-provider/.

[SAML-SecP(v2.0)] FREDERICK HIRSCH, ROB PHILPOTT, und EVE MALER. *Security and Privacy Considerations for the OASIS Security Assertion Markup Language (SAML) V2.0*. OASIS Standard, 15.03.2005. http://docs.oasis-open.org/security/saml/v2.0/saml-sec-consider-2.0-os.pdf, 2005.

[SAML(v2.0)] SCOTT CANTOR, JOHN KEMP, ROB PHILPOTT, und EVE MALER. *Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) V2.0*. OASIS Standard, 15.03.2005. http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf, 2005.

[SGBV] *Sozialgesetzbuch - Fünftes Buch (V) - Gesetzliche Krankenversicherung*. zuletzt geändert durch Art. 1 G v. 30.7.2009 I 2495. http://bundesrecht.juris.de/bundesrecht/sgb_5/, 2009.

[SKS10] PAVOL SOVIS, FLORIAN KOHLAR, und JÖRG SCHWENK. *Security Analysis of OpenID*. In *Proceedings of* Sicherheit 2010, Lecture Notes in Informatics (LNI) (GI-Edition, 2010).

[Slem01] M. SLEMKO. *Microsoft passport to trouble*. http://alive.znep.com/marcs/passport/, 2001.

[TaWa09] RYU WATANABE und TOSHIAKI TANAKA. *Federated Authentication Mechanism using Cellular Phone - Collaboration with OpenID*. In *ITNG '09: Proceedings of the 2009 Sixth International Conference on Information Technology: New Generations*, Seiten 435–442 (IEEE Computer Society, 2009).

[TsTs07] EUGENE TSYRKLEVICH und VLAD TSYRKLEVICH. *Single Sign-On for the Internet: A Security Story*. http://www.orkspace.net/secdocs/Conferences/BlackHat/USA/2007/OpenID%20-%20%20Single%20Sign-On%20for%20the%20Internet-paper.pdf, 2007.

[Wagn09] OLIVER WAGNER. *Eine Milliarde OpenID Accounts*. December 17, 2009. http://www.agenturblog.de/2009-12/eine-milliarde-openid-accounts/.

[Yadis(v1.0)] JOAQUIN MILLER. *Yadis Specification - Version 1.0*. March 18, 2006. http://yadis.org/wiki/Yadis_1.0_%28HTML%29.