Die Open eCard App für mehr Transparenz, Vertrauen und Benutzerfreundlichkeit beim elektronischen Identitätsnachweis

1. Einleitung

Der neue Personalausweis (nPA) setzt einen neuen Meilenstein bei hoheitlichen Dokumenten in Deutschland und Europa. Mit der eID- und eSign-Funktion sowie dem hohen Datenschutzstandard bietet er als Bestandteil der eCard-Strategie eine hervorragende Grundlage und großes Potenzial für vielseitige Anwendungen. Bei der Einführung kam es leider zu technischen und kommunikativen Problemen, die das Vertrauen beschädigt haben. Auch fällt die Anzahl der verfügbaren Anwendungen noch gering aus.

Erforderlich sind daher Impulse und Anreize für Wirtschaft und Forschung, die neuen Technologien zu verwenden und weiter zu entwickeln. Das in diesem Dokument vorgestellte Open eCard Projekt möchte mit der darin entstehenden Open eCard App dazu einen Beitrag liefern.

Das vorliegende Dokument gliedert sich wie folgt: Kapitel 2 enthält eine kritische Würdigung der AusweisApp des Bundes und identifiziert Verbesserungspotenzial. Kapitel 3 stellt das Open eCard Projekt vor und Kapitel 4 beschreibt das Design der Open eCard App sowie den derzeitigen Stand der Entwicklungen. Eine Zusammenfassung und einen Ausblick findet sich schließlich in Kapitel 5.

2. Die AusweisApp des Bundes

Die AusweisApp des Bundes [AusweisApp] wurde zusammen mit dem Start des nPA im November 2010 veröffentlicht und seitdem mehrfach aktualisiert. Inzwischen ist sie in der Version 1.9.0 (Stand Oktober 2012) für die Plattformen Windows, MacOS (bis 10.7) und verschiedene Linux-Derivate kostenfrei erhältlich. Sie zählt zu den am weitesten verbreiteten eID-Anwendungen für den nPA.

Trotz eines aufwändigen Marketingkonzepts ist es der Bundesregierung und den am nPA unmittelbar beteiligten Unternehmen und Institutionen bisher leider nicht gelungen, eine wirklich breite Akzeptanz für die eID-Funktion des Ausweises zu schaffen. Weniger als jeder dritte Bürger, der eine neue Karte erhält, aktiviert auch die eID-Funktion. Gleichzeitig zeigt sich, dass es nur sehr wenige Diensteanbieter gibt. Laut „Kompetenzzentrum neuer Personalausweis“ (www.ccepa.de) sind es derzeit (Stand Oktober 2012) weniger als 50 Anbieter, viele davon Behörden. Von einer guten Aufnahme der eID-Funktionalität durch die Wirtschaft kann daher nicht die Rede sein – zumal die großen Internetdienstanbieter wie eBay, Amazon, Facebook, Google etc. nicht darunter sind. Die Vorstellungen und Wünsche haben sich daher bisher leider nicht erfüllt.

Die aktuelle Situation hat daher deutliches Verbesserungs- und Optimierungspotenzial. Sie könnte zum einen durch neue Zugangsformen, zum anderen durch die Bereitstellung neuer Dienste signifikant verbessert werden. Dazu ist jedoch die AusweisApp in ihrer gegenwärtigen Form nicht in der Lage, verhindert sogar die Erschließung neuer Anwendungsfelder durch Privatleute und nicht unmittelbar beteiligte Unternehmen.

Die immer stärkere Bedeutung von mobilen Geräten und mobilen Anwendungen erfordert eine mobile Auslegung der eCard-Strategie. Insbesondere weil deren Ziele, wie beispielsweise eine sichere und vertrauenswürdige Authentisierung, auch im mobilen Umfeld von hohem Interesse sind. Die NFC-Technologie ermöglicht dabei die kontaktlose Kommunikation mit Chipkarten im mobilen Umfeld. Auch wenn dabei derzeit noch Probleme in Verbindung mit dem nPA auftreten (fehlende Unterstützung von Extended Length APDUs gemäß ISO/IEC 7816 in den Smartphones) und die eGK wegen der kontaktbehafteten Schnittstelle nicht direkt genutzt werden kann, lassen sich bereits Kartenleser mit USB- oder Bluetooth-Schnittstelle mit

mobilen Geräten verbinden. Die Unterstützung mobiler Plattformen gewinnt daher immer weiter an Bedeutung – besonders da es nicht nur allein um Smartphones geht, sondern auch um die immer mehr verbreiteten Tablet-Computer.

Auch im stationären Bereich besteht durch die schleppende Unterstützung von Apples Mac OS X Handlungsbedarf. Gerade dieser Missstand hält eine Reihe namhafter Internet-Unternehmen vom Einsatz der eID-Funktion ab.

Hinzu kommen technische Mängel wie beispielsweise die schleppende Umsetzung aktualisierter Spezifikationen oder die fehlende Unterstützung von TLS 1.1. Dies führt neben Sicherheitsrisiken durch bekannte Angriffe auf TLS 1.0, auch zu einem „zweiten Ökosystem“. Das heißt, die weiteren Teilnehmer der eID-Infrastruktur müssen Anpassungen vornehmen, damit die Interaktion der Komponenten funktioniert.

Weitere Anwendungsbereiche ließen sich eröffnen, wenn auch die eSign-Funktion des nPA oder anderer Chipkarten der eCard-Strategie (eGK, HBA, Bank- und Signaturkarten) nutzbar wäre. Zwar gibt es dafür bereits separate Anwendungen, doch würde es sich für die Nutzer deutlich leichter gestalten, wenn eID- und eSign-Funktion durch eine einzige Anwendung bereitgestellt würden. Dies gilt auch für die weiteren Funktionen der eGK, die derzeit von den Versicherten mangels freier Software-Lösungen – und natürlich mangels einer PIN von ihrer Krankenkasse – überhaupt nicht zur Verfügung stehen. Insbesondere mit Hinblick auf die Akzeptanz der Nutzer ist ein reibungsloser und flächendeckender Nutzungsweg wichtig.

Schließlich ist zu beachten, dass es sich bei der AusweisApp des Bundes zwar um eine kostenlose, aber keineswegs – im Gegensatz zu früheren Ankündigungen – quelloffene Software handelt. Wie bei allen Anwendungen im Bereich der IT-Sicherheit ist eine vertrauenswürdige Sicherheit erst dann gegeben, wenn der Quellcode von unabhängigen Experten überprüft werden kann. Zudem bieten, neben einer stärkeren Vertrauensbasis und einer besseren Modifizierbarkeit, Open-Source-Projekte einen hohen Anreiz und Impulse für Innovation und Forschung. Das große Potenzial der eCard-Strategie könnte mit offenen Standards und offener Software besser unterstützt werden.

3. Das Open eCard Projekt

3.1 Überblick

Im Open eCard Projekt haben sich industrielle und akademische Experten zusammengefunden, um eine quelloffene und plattformunabhängige Implementierung des eCard-API-Frameworks [BSI-TR03112] bereitzustellen, durch die beliebige Anwendungen für Zwecke der Authentisierung und Signatur leicht auf beliebige Chipkarten zugreifen können.

In einer ersten Projektphase soll dieses Rahmenwerk für die Realisierung einer leichtgewichtigen, vertrauenswürdigen und gleichsam gut bedienbaren Alternative zur AusweisApp des Bundes – der in diesem Beitrag vorgestellten Open eCard App (siehe auch [HPS+12]) – genutzt werden.

3.2 Möglichkeiten zur Mitwirkung

Das Open eCard Team ist eine **offene Gemeinschaft**, die alle interessierten Bürgerinnen und Bürger sowie entsprechende Institutionen und Verbände zur Mitwirkung aufruft. Wer die Entwicklung der Open eCard App aktiv unterstützen möchte, kann sich unter http://openecard.org/join registrieren. Wer lediglich über Neuigkeiten aus der Open eCard Community informiert werden möchte, sollte eine E-Mail an subscribe@openecard.org senden. Sonstige Anregungen nimmt das Open eCard Team gerne unter feedback@openecard.org entgegen.

3.3 Qualitätsmanagement

3.3.1 Besondere Problematik

Die Open eCard App wird für verschiedene Plattformen ausgeliefert und in unterschiedlichsten Kontexten durch Menschen mit unterschiedlichster IT-Prägung verwendet. Die Qualität elektronischer Identitätsdokumente ist nicht auf Sicherheit im Sinne von IT-Sicherheit zu reduzieren: Die Qualität erfordert die Betrachtung der Ebenen Semantik/Usability, Code-Sicherheit und kryptografische Verfahren.

In der Literatur ist hinlänglich auf die subjektive Sicherheit als Akzeptanzfaktor hingewiesen worden. Die Akzeptanz des nPA ist neben den klassischen Kriterien wie „gefühlter Vorteil gegenüber der nächstschlechteren Lösung“, „geringe Komplexität“, „Ausprobierbarkeit“, „Sichtbarkeit“ etc., vgl. [Rogers03] essentiell von der subjektiven Sicherheit der Verwendung abhängig. Jeder in den Medien dokumentierte Fall eines Missbrauchs der eID-Funktion oder – noch schlimmer – einer Signatur bedeutet schwindende Akzeptanz.

Die Verfügbarkeit des Quellcodes ermöglicht es, die Integrität der kompilierten Software zu prüfen. Damit wird ein bisher unerreichtes Vertrauensniveau erreicht. Zugleich kann der Quellcode missbraucht werden, um eine täuschend echte AusweisApp zu konstruieren, die im Hintergrund ungewollte Aktionen auslöst.

Viele weitere Aspekte führen zu dem Schluss, dass Qualitätssicherung auf mehreren Ebenen besonderer Anstrengungen bedarf. Im Folgenden sind Methoden und Organisation des Open eCard App Projektes dargelegt, die den Bedrohungen entgegenwirken.

3.3.2 Methodik

Um auf der Ebene der Semantik/Usability erfolgreich zu sein, muss eine qualitative Analyse mit unterschiedlichen Nutzergruppen bei vorhandenen Dienstanbietern erfolgen. Die Kommunikationsmuster, die „Ansprache“ des Nutzers, der Informationsaustausch sind hier entscheidend. Folglich ist es Aufgabe eines Usability-Expertenteams im Projekt, einen repräsentativen Querschnitt an Nutzern in unterschiedlichen Anwendungsfällen zur Mitarbeit bei der Bewertung der Releases zu gewinnen. Ein öffentliches Zielkundenprofil, ein Bewertungsprotokoll einschließlich der Tests und Ergebnisse sind Resultate dieser Aktivitäten.

Die Sicherheit des verwendeten Codes wird durch bestehende Best-Practices im Open Source erreicht. Dazu zählen

- Programmierkonventionen einschließlich wirksamer Mechanismen.
- Trennung der Verantwortlichkeiten: Entwickler erstellen Code, den besonders erfahrene Entwickler (Committer) prüfen und in das Repository übernehmen.
- Regelmäßige statische Quellcode-Analyse und daran anschließende Bugfixes. Diese ermöglichen das objektive und kontinuierliche Messen der Reife.
- Unit-Tests und Mittel, die Testabdeckung zu prüfen
- sowie ein für die kontinuierliche Integration geeigneter Build-Prozess.

Die Summe dieser Maßnahmen erreicht Werte von 0,1 Fehlern pro 1000 Zeilen Quellcode, was vergleichbar bzw. besser ist, als kommerzielle Software (vgl. [Cov11]). Der Schlüssel liegt hier im kontinuierlichen Verbesserungsprozess.

Die korrekte Anwendung kryptografischer Protokolle und Mechanismen wird durch Peer-Review erreicht. Dies erfordert, wie auch bei der Entwicklung, hochspezialisierte Informatiker, deren kritisches Hinterfragen der Implementierung durch nichts zu ersetzen ist. Der Nutzen des Peer-Reviews ist umso größer, je ernsthafter der Reviewer versucht, Lücken zu finden. Dies wird durch sog. „Hostile-Review“ erreicht.

3.3.3 Organisation

Es wird ein Security-Team etabliert, das folgende Aufgaben wahrnimmt:

- Entgegennehmen, Verarbeiten und Schließen von gemeldeten Sicherheitslücken
- Hilfestellung für die Maintainer von Modulen bei der Beseitigung von Sicherheitslücken
- Erstellung und Pflege der Programmierkonventionen
- Erstellen von Dokumentation zu sicheren Integrationsmöglichkeiten

Das Team rekrutiert sich aus anerkannten Entwicklern und Beratungshäusern.

4. Die Open eCard App

4.1 Grundlegendes Design

Vor dem Hintergrund der existierenden Spezifikation des eCard-API-Framework [BSI-TR03112], der in [HPS+12] näher erläuterten Anforderungen und den im Zuge der verschiedenen Implementierungen [AusweisApp], [bos-Autent], [Ageto-AA], [OpenPACE], [Petr11], [Hors09] und [Hors11] gesammelten Erfahrungen wurde der in Abbildung 1 dargestellte Architektur-Entwurf für die Open eCard App entwickelt. Durch den hochgradig modularen Ansatz und die plattformunabhängige, Java-basierte Realisierung der Kernmodule kann die Open eCard App leicht erweitert und auf unterschiedlichen Plattformen (z.B. Windows, Linux, Mac OS, Android etc.) eingesetzt werden.

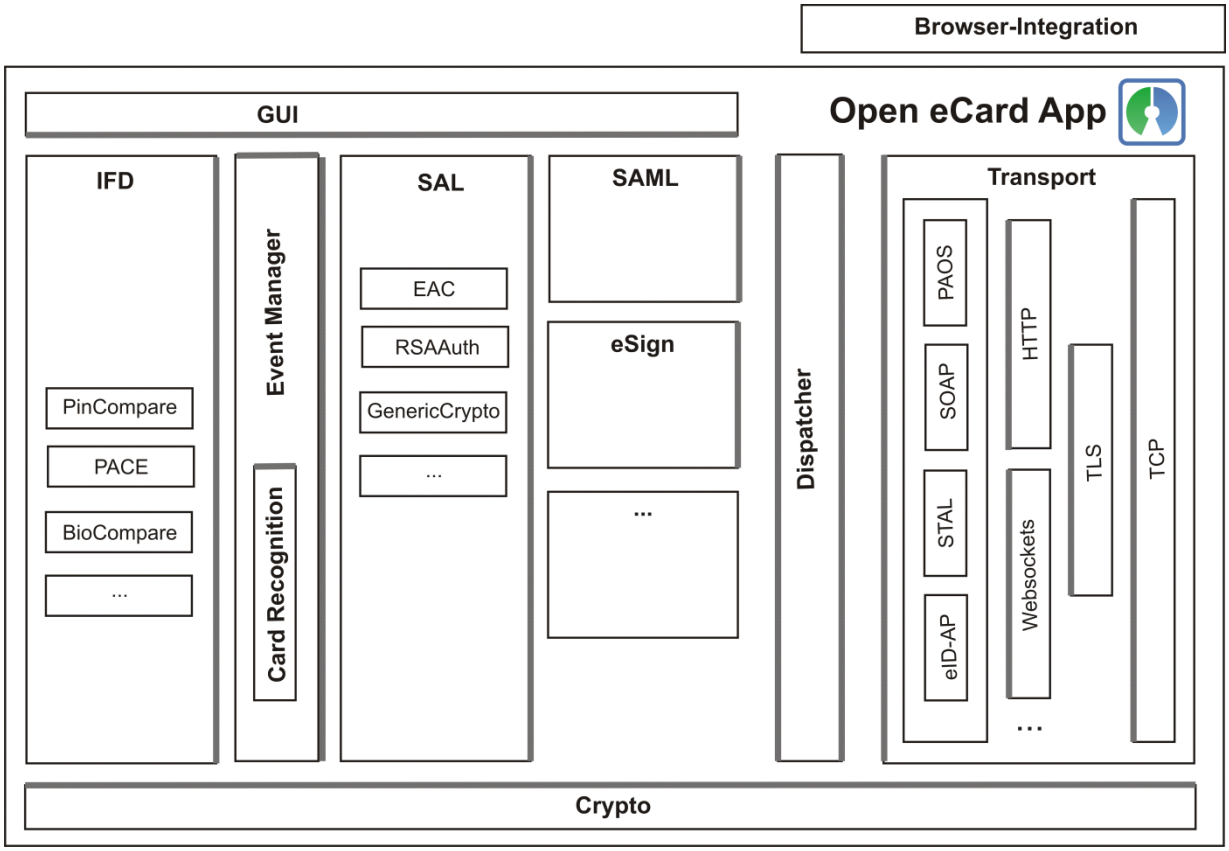

Abbildung 1: Die Architektur der Open eCard App

Die wesentlichen Module der Open eCard App sind im Folgenden näher erläutert:

- Interface Device (IFD)

  Diese Komponente implementiert das IFD-Interface, das in Teil 6 von [BSI-TR03112] und Teil 4 von [ISO24727] spezifiziert ist. Weiterhin enthält es zusätzliche Schnittstellen für Passwort-basierte Protokolle (z.B. PACE) und stellt eine einheitliche Schnittstelle für die Kommunikation mit unterschiedlichen Kartenlesern und Smartcards zur Verfügung.

- Event Manager

  Der Event Manager überwacht eintreffende Events (z.B. das Hinzufügen oder Entfernen von Lesegeräten oder Karten) und führt die Typerkennung von neu hinzugefügten Karten durch. Da für die Erkennung des Chipkartentyps die in [CEN15480] bzw. [ISO24727] (Amd1) spezifizierten CardInfo-Dateien genutzt werden, kann die Open eCard App sehr leicht zusätzliche Chipkarten unterstützen. Die derzeit unterstützten Chipkarten sind unter http://openecard.org dargestellt und ein Demonstrator für eine intuitiv benutzbare Web-Applikation auf Basis der Open eCard App findet sich unter http://openecard.org/demo.

- Service Access Layer (SAL)

  Dieses Modul implementiert den Service Access Layer wie er in Teil 4 von [BSI-TR03112] und Teil 3 von [ISO24727] spezifiziert ist. Ein wichtiger Aspekt dieser Komponente ist sein Erweiterungsmechanismus, welcher das Hinzufügen neuer Authentisierungsprotokolle ermöglicht, ohne andere Teile der Implementierung ändern zu müssen.

- Crypto

  Das Crypto-Modul vereinheitlicht den Zugriff auf kryptographische Funktionen für andere Module. Durch die Nutzung der Java Cryptography Architecture (JCA) [JCA] und der entsprechenden Provider-Architektur kann der Cryptographic Service Provider (CSP) leicht ausgetauscht werden.

- Graphische Benutzerschnittstelle (GUI)

  Die GUI wird über eine abstrakte Schnittstelle angesprochen und ist daher jederzeit leicht gegen eine andere Implementierung austauschbar. Dies ermöglicht Plattform-spezifische GUI Implementierungen, ohne andere Module ändern zu müssen.

- Security Assertion Markup Language (SAML)

  Diese Komponente bietet Unterstützung für erweiterte SAML-Profile, wie z.B. [SAML-ECP], und Bindings, wie z.B. [SAML-HoK], die zu einem effizienteren und besser geschützten Authentisierungsablauf führen.

- Elektronische Signaturen (eSign)

  Diese Komponente ermöglicht die Erzeugung fortgeschrittener elektronischen Signaturen nach [ETSI-101733], [ETSI-101903] und [ETSI-102778]. Die Komponente implementiert eine an Teil 2 von [BSI-TR03112] angelehnte und auf [OASIS-DSS] basierende Schnittstelle.

- Dispatcher

  Der Dispatcher stellt eine zentrale Komponente dar, die alle eintreffenden und ausgehenden Nachrichten an die entsprechenden Module weiterleitet. Durch diese Art der Zentralisierung kann die Gesamtkomplexität der Open eCard App deutlich reduziert werden.

- Transport

  Diese Komponente kapselt die individuellen Transportprotokolle auf den unterschiedlichen Schichten. Durch die eingesetzte Architektur ist es leicht, die verschiedenen Protokolle zu nutzen und weitere hinzuzufügen. Um die momentan vorhandenen eID-Server bedienen zu können, unterstützt die Open eCard App den Austausch von PAOS-basierten Nachrichten per HTTP und die gesicherte Übertragung mit TLS. Der Protokoll-Stack ist aber bewusst so entworfen, dass auch beliebige andere Bindings (z.B. SOAP [SOAP-v1.1]) und alternative Protokolle wie der österreichische Security Token Abstraction Layer (STAL) [MOCCA] oder das belgische eID Applet Protokoll [eID-Applet] unterstützt werden können.

- Browser Integration

  Damit die Open eCard App nicht nur in Verbindung mit fest installierten Fachanwendungen, sondern auch in Browser-gestützten Webanwendungen genutzt werden kann, wird der in Teil 7 von [BSI-TR03112] spezifizierte, HTTP-basierte eID-Aktivierungsmechanismus unterstützt. Längerfristig ist die tiefere Integration der Open eCard App in populäre Browser über die von diesen unterstützten kryptografischen Schnittstellen (z.B. [PKCS#11]) anvisiert.

4.2 Ausgewählte Anwendungsfälle

Die Vorteile einer Zwei-Faktor-Authentisierung, beispielsweise durch eine Chipkarte und der dazugehörigen PIN, sind bekannt. Neben einer reinen Fokussierung auf eine Online-Authentisierung durch den nPA, ist es denkbar, die Open eCard App stärker in das Betriebssystem zu intergieren und so beispielsweise auf das unter Linux genutzte PAM (Pluggable Authentication Modules) bzw. den Microsoft-spezifischen Login-Mechanismus für Windows zurückzugreifen. Das Ziel ist hierbei eine chipkartenbasierte Anmeldung am Betriebssystem mittels der Open eCard App.

Insbesondere die Integration und Interaktion mit existierenden Infrastrukturen und Projekten ist ein wichtiger Punkt, um die Verbreitung und Akzeptanz zu stärken. Dazu zählt beispielsweise die Interaktion mit GnuPG, um das Signieren und Verschlüsseln von Mails mittels der Open eCard App für beliebige Chipkarten zu ermöglichen. Denkbar ist auch die Interaktion mit TrueCrypt, um die Passwörter zur Datenverschlüsselung oder die „Keyfiles“ auf einer Chipkarte zu speichern.

Wünschenswert ist es ebenfalls, das der Bund, wie beispielsweise beim Kraftfahrt-Bundesamt geschehen, die eID-Infrastruktur weiter in die eigenen Progresse intergiert. Beispielsweise könnte die Authentisierung, Signatur und Verschlüsselung bei der eVergabe, um die eID-Infrastruktur erweitert werden. Durch die Schaffung neuer Einsatzmöglichkeiten könnte der Bund die eigene eCard-Strategie stärken.

4.3 Stand der Entwicklung

Die Open eCard App basiert auf dem in Kapitel 4.1 vorgestellten Design. Die App unterstützt bereits eine Kartenerkennung auf Basis von CardInfo-Dateien, eine Authentisierung via eID-Funktion mit dem nPA und eine prototypische zertifikatsbasierte TLS-Client-Authentisierung, die beispielsweise mit der eGK oder den durch CardInfo-Dateien beschriebenen Signaturkarten genutzt werden kann. Die Open eCard App wird dabei in folgenden Ausführungen bereitgestellt:

4.3.1 Browser Applet

Das Java Applet ist als Demonstrator unter http://openecard.org/demo verfügbar.

4.3.2 Rich Client

Der Rich Client ist als dauerhaft installierte Anwendung für die stationären Betriebssysteme Windows, Mac OS X und Linux konzipiert. Für diese Betriebssysteme stellt die Open eCard App einen Installer bereit. Zusätzlich ist eine portable Installation[1] in der Entwicklung.

4.3.3 Android

Die Android App befindet sich auf einem mit den stationären Lösungen vergleichbaren Stand. Dies beinhaltet insbesondere den automatischen Start der Anwendung via Link (Localhost und TCToken) aus dem Browser heraus, sowie die Durchführung einer Authentisierung mittels nPA oder einer zertifikatsbasierten TLS-Client-Authentisierung mit der eGK.

Durch die NFC-Problematik und der kontaktbehafteten Schnittstelle der eGK wird die Kommunikation mit den Karten zurzeit über einen per USB angeschlossenen Kartenleser realisiert. Die Einbindung der Kartenleser erfolgt über PCSC-lite und setzt „gerootete“ Geräte voraus.

5. Zusammenfassung und Ausblick

Die Open eCard App bietet durch ihr fundiertes und an etablierten Standards ausgerichtetes Design eine hervorragende und robuste Grundlage für die Entwicklung einer universellen, modularen und benutzerfreundlichen eID-Applikation. Weitere Protokolle können bequem auf den passenden Ebenen wie IFD, SAL oder Transport hinzugefügt werden und neue Chipkarten können leicht anhand der dazugehörigen CardInfo-Dateien integriert werden.

Die weiteren Entwicklungen der Open eCard App betreffen insbesondere die qualitätsgesicherte Integration der TLS-basierten Authentisierung, einer erweiterten Internationalisierung sowie der Unterstützung der Signaturfunktion. Des Weiteren ist in der Open eCard App eine effiziente und sichere SAML-Unterstützung sowie längerfristig die Unterstützung Attribut-basierter Credentials geplant.

Literatur

[Ageto-AA] Ageto Innovation GmbH: *AGETO AusweisApp*, http://www.ageto.de/egovernment/ageto-ausweis-app

[AusweisApp] BSI: Offizielles Portal für die *„AusweisApp“*, http://www.ausweisapp.de

[bos-Autent] bos GmbH & Co. KG: *Governikus Autent*, http://www.bos-bremen.de/de/governikus_autent/1854605/

[BSI-TR03112] BSI: *eCard-API-Framework*, Technical Directive of the Federal Office for Information Security Nr. 03112, BSI TR-03112, Version 1.1, http://docs.ecsec.de/BSI-TR-03112

[CEN15480] CEN: *Identification card systems — European Citizen Card*, CEN TS 15480 (Part 1-4)

[eID-Applet] F. Cornelis & al.: *eID-Applet Project*, http://code.google.com/p/eid-applet/

[ETSI-101733] ETSI: *CMS Advanced Electronic Signatures (CAdES)*, ETSI TS 101 733, Version 1.8.1, December 2009

[ETSI-101903] ETSI: *Technical Specification XML Advanced Electronic Signatures (XAdES)*, ETSI TS 101 903, Version 1.4.1, June 2009

[ETSI-102778] ETSI: *PDF Advanced Electronic Signature Profiles*, ETSI TS 102 778, part 1-5, 2009

[Hors09] M. Horsch: *MobilePACE – Password Authenticated Connection Establishment implementation on mobile devices*, Bachelor Thesis, TU Darmstadt, 2009, http://www.cdc.informatik.tu-darmstadt.de/mona/pubs/200909_BA_MobilePACE.pdf

[Hors11] M. Horsch: *MONA - Mobile Authentication with the new German eID-card (in German)*, Master Thesis, TU Darmstadt, 2011, http://www.cdc.informatik.tu-darmstadt.de/mona/pubs/201107_MA_Mobile%20Authentisierung%20mit%20dem%20neuen%20Personalausweis%20(MONA).pdf

[HoSt11] M. Horsch, M. Stopczynski: *The German eCard-Strategy*, Technical Report: TI-11/01, TU Darmstadt, http://www.cdc.informatik.tu-darmstadt.de/reports/reports/the_german_ecard-strategy.pdf

[1] Gemäß der von http://portableapps.com bereitgestellten Infrastruktur.

[HPS+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*, Sicherheit 2012, GI LNI, 2012, http://subs.emis.de/LNI/Proceedings/Proceedings195/95.pdf

[ISO24727] ISO/IEC: *Identification cards – Integrated circuit cards programming interfaces*, ISO/IEC 24727 (Part 1-5)

[JCA] Oracle: *Java ™ Cryptography Architecture (JCA) Reference Guide*, http://download.oracle.com/javase/6/docs/technotes/guides/security/crypto/CryptoSpec.html

[Kowa07] B. Kowalski: *Die eCard-Strategie der Bundesregierung*, in BIOSIG 2007: Biometrics and Electronic Signatures, Proceedings of the Special Interest Group on Biometrics and Electronic Signatures, LNI 108, pp. 87–96, 2007, http://subs.emis.de/LNI/Proceedings/Proceedings108/gi-proc-108-008.pdf

[MOCCA] MOCCA: *Modular Open Citizen Card Architecture Project*, http://mocca.egovlabs.gv.at/

[OASIS-DSS] OASIS: *Digital Signature Service Core Protocols, Elements, and Bindings*, Version 1.0, OASIS Standard, via http://docs.oasis-open.org/dss/v1.0/oasis-dss-core-spec-v1.0-os.pdf

[OpenPACE] F. Morgner & al.: *OpenPACE Project - Crypto library for the PACE protocol,* http://openpace.sourceforge.net/

[PAOS-v2.0] Liberty Alliance Project: *Liberty Reverse HTTP Binding for SOAP Specification*, Version v2.0, via http://www.projectliberty.org/liberty/content/download/909/6303/file/liberty-paos-v2.0.pdf

[Petr11] D. Petrautzki: *Security of Authentication Procedures for Mobile Devices*, Master Thesis, Hochschule Coburg, 2011

[PKCS#11] RSA Laboratories: *PKCS #11 Base Functionality v2.30: Cryptoki – Draft 4*, 10 July 2009

[SAML-ECP] S. Cantor & al.: *SAML V2.0 Enhanced Client or Proxy Profile Version 2.0*, Working Draft 02, 19.02.2011, http://www.oasis-open.org/committees/download.php/41209/sstc-saml-ecp-v2.0-wd02.pdf

[SAML-HoK] N. Klingenstein, T. Scavo: *SAML V2.0 Holder-of-Key Web Browser SSO Profile Version 1.0*, Committee Specification 02, 10.08.2010, http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-holder-of-key-browser-sso.pdf

[SOAP-v1.1] W3C Note: *Simple Object Access Protocol (SOAP) 1.1*, 8 May 2000, via http://www.w3.org/TR/2000/NOTE-SOAP-20000508

[Rogers03] E. M. Rogers und E. Rogers, *Diffusion of Innovations, 5th Edition*. Free Press, 2003.

[Cov11] „Coverity Scan: 2011 Open Source Integrity Report“, http://www.coverity.com/library/pdf/coverity-scan-2011-open-source-integrity-report.pdf