Moritz Horsch, Detlef Hühnlein, Anja Lehmann, Johannes Schmölz und Tobias Wich

Authentisierung mit der Open eCard App

Von Ausweiskarten zu datenschutzfreundlichen Credentials

Für die starke Authentisierung und den elektronischen Identitätsnachweis stehen unterschiedliche Mechanismen zur Verfügung. Aus dem Blickwinkel des Datenschutzes reicht die Bandbreite von der Authentisierung mit X.509-Zertifikaten über das vom neuen Personalausweis unterstützte Extended Access Control Protokoll bis hin zu datenschutzfreundlichen Credentials. Der vorliegende Beitrag erläutert, wie diese unterschiedlichen Technologien mit der Open eCard App genutzt werden können.

Einleitung

Die zahlreichen Schlagzeilen über Schwachstellen in Cloud- und Webanwendungen, die latente Schwäche der passwort-basierten Authentifizierung [1] und der damit oft verbundene Identitätsmissbrauch [2] unterstreichen den Bedarf an starken, auf mehreren Faktoren basierenden, Authentisierungsmechanismen. Viele US-amerikanische Cloud-Provider unterstützen inzwischen ergänzende Einmalpasswort-Verfahren (vgl. [3], [4], [5]). In der FIDO-Alliance [6] haben sich namhafte Unternehmen zusammengeschlossen, um der passwort-basierten Authentifizierung den Kampf anzusagen. In Europa bietet es sich an, die bereits ausgegebenen chipkarten-basierten Ausweise für eine starke Authentisierung zu nutzen [7], da hierdurch neben der Authentisierung auch ein elektronischer Identitätsnachweis realisiert werden kann.

Betrachtet man die verschiedenen in Europa eingesetzten elektronischen Ausweise [9] und ihre unterstützten Authentisierungsprotokolle, so erscheint neben dem Extended Access Control (EAC) Protokoll [10], wie es beim neuen Personalausweis (nPA) eingesetzt wird, vor allem das TLS-Protokoll [11] relevant, da die bislang häufig auf Ausweiskarten aufgebrachten X.509-Zertifikate direkt für die Client-Authentisierung in Webanwendungen genutzt werden können. Enthalten die X.509-basierten Zertifikate personenbezogene Daten (wie z.B. in Estland [12] und Finnland [13]), so müssen diese Daten zur Authentisierung stets vollständig übermittelt werden, da ansonsten das Zertifikat nicht verifiziert werden kann. Desweiteren fungieren das Zertifikat selbst und der darin enthaltene öffentliche Schlüssel des Nutzers als eindeutige Identifikatoren mit denen sämtliche Authentisierungen der Nutzer miteinander verknüpft werden können. Insbesondere wenn die Authentisierung über einen unverschlüsselten Kanal erfolgt, ist diese X.509- und TLS-basierte Authentisierung aus dem Blickwinkel des Datenschutzes nicht optimal. Mit Hilfe attribut-basierter und datenschutzfreundlicher Credentials [14], [15], [16], [17], [18] kann eine ebenso starke Authentisierung wie mit X.509-Zertifikaten erreicht werden, die aber gleichzeitig einen besonders sparsamen Umgang mit persönlichen Daten erlaubt. Datenschutzfreundliche Credentials erlauben eine selektive Auswahl der Attribute, die in einer bestimmten Authentisierungssession übermittelt werden und hinterlassen bei der Nutzung auch keine eindeutigen Identifikatoren mit denen verschiedene Sessions verknüpft werden können.
Der vorliegende Beitrag beschreibt, wie diese technologisch sehr unterschiedlichen Authentisierungsmechanismen in der leichtgewichtigen und als Open Source verfügbaren Open eCard App [19], [20] genutzt werden können. Hierfür wird in Abschnitt 1 die modulare und erweiterbare Architektur der Open eCard App vorgestellt und in Abschnitt 2 näher auf die Umsetzung der unterschiedlichen Authentisierungsmechanismen eingegangen.

1. Architektur der Open eCard App

Die Open eCard App [19], [20] ist eine, auf dem eCard-API-Framework basierende, Open Source Client-Software, die die Nutzung unterschiedlicher Chipkarten über den in CEN 15480 [25] bzw. ISO/IEC 24727 [32] definierten CardInfo-Mechanismus ermöglicht. Durch den modularen Aufbau der Open eCard App und den Erweiterungsmechanismus kann die Funktionalität leicht geändert oder ergänzt werden. Die Architektur der Open eCard App ist in Abbildung 1 dargestellt.

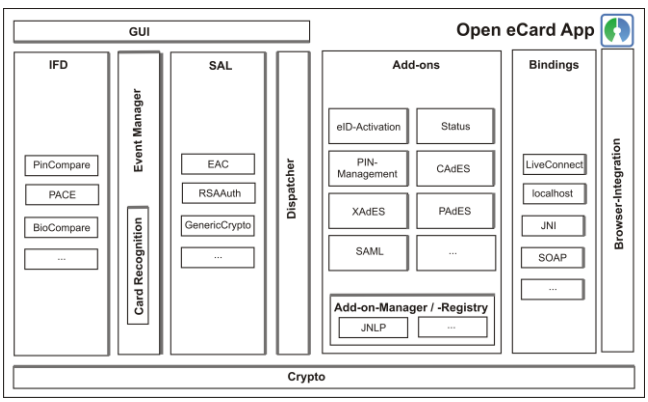

Abbildung 1: Modulare Architektur der Open eCard App

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

- Interface Device (IFD)
  Dieses Modul implementiert die in TR-03112-6 [32] und ISO/IEC 24727-4 [33] spezifizierte IFD-API und bietet zusätzliche Schnittstellen für passwort-basierte Protokolle, wie z.B. PACE [10]. Es ermöglicht den Zugriff auf unterschiedliche Kartenlesegeräte und Smartcards über eine einheitliche Schnittstelle.
- Event Manager
  Der Event Manager überwacht den Zustand des IFD und benachrichtigt andere Komponenten, die sich zuvor bei ihm registriert haben, über Veränderungen, wie z.B. das Hinzufügen eines Kartenlesegerätes oder das Ziehen einer Karte. Außerdem erfolgt hier die Erkennung des Kartentyps über den in CEN 15480 [25] bzw. ISO/IEC 24727 [33] standardisierten CardInfo-Mechanismus.
- Service Access Layer (SAL)
  Dieses Modul implementiert den SAL, der in Teil 4 der TR-03112 [32] sowie in Teil 3 von ISO/IEC 24727 [33] spezifiziert ist. Neben dem Zugriff auf Smartcards, der mittels CardInfo-Dateien gesteuert wird, und der Verwaltung von Kartenapplikationen, ist der SAL insbesondere für die Durchführung unterschiedlicher Authentisierungsprotokolle zuständig. Über den unten und in [34] näher erläuterten Plugin-Mechanismus können weitere Authentisierungsprotokolle ohne Anpassungen am SAL hinzugefügt werden.
- Dispatcher
  Der Dispatcher verteilt alle ein- und ausgehenden Nachrichten an die entsprechenden Module. Durch diese Zentralisierung wird zum Einen die Komplexität des Gesamtsystems reduziert und zum anderen kann zusätzliche Logik zur Steuerung des Datenflusses eingebracht werden. Hierdurch lassen sich beispielsweise unterschiedliche Nachrichten-Filter realisieren.
- Add-ons
  Über den in Abschnitt 2.1 und in [34] näher erläuterten Erweiterungsmechanismus kann die Open eCard App mit zusätzlicher Funktionalität versehen werden.
- Bindings
  Dieses Modul enthält unterschiedliche Bindings, die für die Kommunikation zwischen der Open eCard App und weiteren Anwendungen zuständig ist. Beispielsweise kann eine eigenständige Open eCard App mit dem localhost-Binding im Stile von [32] (Teil 7, Abschnitt 3.2.1) oder ein Open eCard Applet mit dem LiveConnect-Binding [35] angesprochen werden. Durch ein Binding-Modul werden die transportspezifischen Aufrufe in eine kanonische, für die interne Weiterverarbeitung geeignete Form umgewandelt.
- Browser-Integration
  Dieses Modul soll zukünftig für die Integration der Open eCard App in die verschiedenen Browser mittels PKCS#11 [36] oder CSP [37] sorgen.
- Graphische Benutzerschnittstelle (GUI)
  Die GUI wird über eine abstrakte Schnittstelle angesprochen, die unterschiedliche Implementierungen ermöglicht. So ist neben der Swing- oder Android-spezifischen GUI, auch eine Implementierung in JavaScript denkbar. Auch ein „Headless-Mode“ könnte über diese Schnittstelle umgesetzt werden.
- Crypto
  Dieses Modul stellt den anderen Modulen die notwendigen kryptographischen Funktionen inklusive TLS bereit. Die Grundlage für das Crypto-Modul der Open eCard App bildet BouncyCastle [38].

2. Authentisierung mit der eCard App

Die für die Authentisierung relevante Funktionalität der Open eCard App befindet sich im SAL und den für die Initialisierung des Authentisierungsvorganges zuständigen Plugin-Modulen (z.B. eID-Aktivierung).

2.1 Erweiterbarkeit der Open eCard App

2.1.1 IFD- und SAL-Protokolle

Der SAL stellt das generische „Service Access Interface“ gemäß Teil 3 von ISO/IEC 24727 [32] bereit und kann über den nachfolgend und in [34] näher beschriebenen Erweiterungsmechanismus beliebige Authentisierungsprotokolle unterstützen. Insbesondere existieren sowohl im IFD- als auch im SAL-Modul Funktionen, die durch protokollspezifische Identifikatoren parametrisiert werden können. Beispielsweise sind sowohl die kryptographischen Funktionen[1] als auch die Funktionen zur Verwaltung und Nutzung des Schlüsselmaterials[2] – der so genannten Differential Identities (DID) – im SAL abhängig vom Authentisierungsprotokoll. Hierdurch kann beispielsweise im IFD das PACE-Protokoll bzw. im SAL das EAC-, Generic Cryptography- oder PIN Compare-Protokoll (siehe [32], Teil 7) unterstützt werden. Beispielsweise verwendet der neue Personalausweis das PACE- und EAC-Protokoll, während das Generic Cryptography- und PIN Compare-Protokoll bei vielen anderen Signatur- und Ausweiskarten eingesetzt wird.

2.1.2 Add-ons

Auch oberhalb des SAL (bzw. in Abbildung 1 rechts davon) können Erweiterungen über so genannte Add-ons vorgenommen werden. Bei den Add-ons wird zwischen Plug-ins, die von einer externen Anwendung über eine Schnittstelle aufgerufen werden, und eigenständigen Extensions unterschieden. Für die Verwaltung dieser Erweiterungsmodule existiert ein Add-on-Manager und eine Add-on-Registry. Über die Add-on-Registry kann bestimmt werden, welches Add-on für einen bestimmten Vorgang benötigt wird. Außerdem kümmert sich die Add-on-Registry um die Beschaffung des entsprechenden Erweiterungsmoduls. Beispielsweise kann eine solche Registry auf Basis von JNLP [39] realisiert werden. In dieser Variante, die sowohl bei der Applet-basierten als auch bei der per Java Web Start ausgelieferten Open eCard App genutzt wird, ist die Registrierung der Add-ons inhärent vorhanden. Die eigentlichen Erweiterungsmodule werden über den in JNLP vorhandenen Mechanismus bedarfsgerecht nachgeladen und für den schnellen Zugriff lokal vorgehalten. Längerfristig wären alternative Registrierungsmechanismen im Stile eines App Stores

[1] Siehe [32], Teil 4, Abschnitt 3.5.
[2] Siehe [32], Teil 4, Abschnitt 3.6.

denkbar, die bei einer fest installierten Open eCard App zum Tragen kommen könnten.

Nach dem Laden des Erweiterungsmoduls durch die Add-on-Registry übernimmt der Add-on-Manager die Verwaltung der Add-on-Instanzen und kümmert sich um die Einhaltung von Sicherheitsrichtlinien durch einen Sandbox-Mechanismus.

Jedes Plugin stellt eine Beschreibung der von ihm angebotenen Schnittstelle bereit, die mehrere Operationen beinhalten kann. Durch diese abstrakte Schnittstelle ist es möglich, das Binding - also die konkret an einen Transportmechanismus gebundene Schnittstelle zur Außenwelt - auszutauschen oder sogar mehrere Bindings zur Verfügung zu stellen. Bei Extensions hingegen wird die grafische Repräsentation der Applikation, z.B. einem Einstellungsdialog, beschrieben, so dass diese Aktionen automatisiert angezeigt werden können.

2.2 Authentisierungsmodule

Mit dem oben erläuterten Erweiterungsmechanismus kann die Open eCard App um beliebige Funktionalität erweitert werden. Insbesondere können hierdurch verschiedene Authentisierungsmodule für die Unterstützung der unterschiedlichen in Europa ausgegebenen Ausweiskarten bereitgestellt werden.

2.2.1 EAC und PACE für den neuen Personalausweis

Für die Unterstützung des neuen Personalausweises wird das PACE-Protokoll im IFD, das EAC-Protokoll gemäß [32] (Teil 7, Abschnitt 4.6) im SAL und geeignete Mechanismen zur eID-Aktivierung[3] über einen in eine Webseite eingebetteten Link bzw. ein `<object>` Element benötigt. Diese Funktionalität ist in der Open eCard App bereits vollständig umgesetzt und mit den unter http://openecard.org/dienste aufgeführten Anwendungen erfolgreich getestet.

2.2.2 TLS, Generic Crypto und PIN Compare

Für die Unterstützung von Ausweis- und Signaturkarten, die über X.509-Zertifikate verfügen, wird neben der grundlegenden TLS-Funktionalität aus dem Crypto-Modul und einer entsprechenden CardInfo-Datei im SAL das Generic Cryptography-Protokoll gemäß [32] (Teil 7, Abschnitt 4.9) und zumeist das PIN Compare-Protokoll für die Benutzer authentisierung [32] (Teil 7, Abschnitt 4.1) benötigt.

2.2.3 Datenschutzfreundliche Credentials

Durch die offene auf ISO/IEC 24727 basierende Architektur können grundsätzlich auch innovative Protokolle für die Erzeugung und Verwendung von datenschutzfreundlichen Credentials unterstützt werden. Wie in der auf [40] und [41] basierenden Abbildung 2 angedeutet, kann mittels `DIDCreate` ein datenschutzfreundliches Credential erzeugt und mit `DIDAuthenticate` für den datenschutzfreundlichen Identitätsnachweis genutzt werden. Die konkrete Integration attribut-basierter und datenschutzfreundlicher Credentials in den eID-Kontext und die Open eCard App wird zur Zeit in den EU-geförderten Projekten ABC4Trust[4] und FutureID[5] untersucht und in Pilotprojekten[6] in Schweden und Griechenland erprobt.

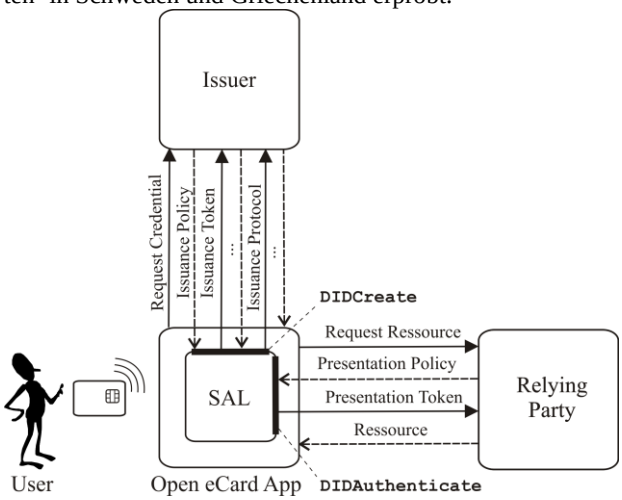

Abbildung 2: Ausstellung und Nutzung von datenschutzfreundlichen Credentials mit der Open eCard App

Fazit

Wie eingangs erläutert, unterstützen die in Europa eingeführten Ausweiskarten unterschiedliche Authentisierungsprotokolle, wobei neben dem vom Personalausweis unterstützten EAC-Protokoll [10] insbesondere das TLS-Protokoll [11] relevant erscheint, da dieses bei Webanwendungen weit verbreitet ist und in Verbindung mit den häufig auf Signatur- und Ausweiskarten vorhandenen X.509-Zertifikaten verwendet werden kann. Zwar erreichen diese Verfahren eine sehr starke Authentisierung, jedoch sind sie aus Sicht des Datenschutzes nicht optimal. Alternative Verfahren wie attribut-basierte, datenschutzfreundliche Credentials ermöglichen hingegen sowohl starke Authentisierung als auch einen sparsamen Umgang mit persönlichen Daten. Die Open eCard App strebt daher eine Unterstützung sowohl der etablierten Verfahren also auch von innovativen, datenschutzfreundlicheren Technologien an.

Literaturverzeichnis

[1] Deloitte, *P@$$1234: the end of strong password-only security,* http://www.deloitte.com/view/en_GX/global/industries/technology-media-telecommunications/tmt-predictions-2013/tmt-predictions-2013-technology/9eb6f4efcbccb310VgnVCM1000003256f70aRCRD.htm#.UXv0YXfaioA, 2013.

[2] Arbeitsgruppe Identitätsschutz im Internet (a-i3), *Webseite,* https://www.a-i3.org.

[3] Google Inc., *Advanced sign-in security for your Google account,* http://googleblog.blogspot.de/2011/02/advanced-sign-in-security-for-your.html, 2011.

[4] Amazon Inc., *AWS Multi-Factor Authentication,*

[3] Siehe http://openecard.org/eID-Aktivierung .

[4] Siehe http://abc4trust.eu .
[5] Siehe http://futureid.eu .
[6] Siehe https://abc4trust.eu/index.php/home/pilots .

http://aws.amazon.com/de/mfa/, 2013.

[5] Microsoft Inc., *Microsoft Account Gets More Secure,* http://blogs.technet.com/b/microsoft_blog/archive/2013/04/17/microsoft-account-gets-more-secure.aspx, 2013.

[6] FIDO Alliance, *Webseite,* http://www.fidoalliance.org/, 2013.

[7] D. Hühnlein, G. Hornung, H. Rossnagel, J. Schmölz, T. Wich und J. Zibuschka, *SkIDentity - Vertrauenswürdige Identitäten für die Cloud,* http://www.ecsec.de/pub/2011_DACH_SkIDentity.pdf, 2011.

[8] D. Hühnlein, J. Schmölz, T. Wich und M. Horsch, „Sicherheitsaspekte beim chipkartenbasierten Identitätsnachweis," in *Daten- und Identitätsschutz in Cloud Computing, E-Government und E-Commerce*, Georg Borges, Jörg Schwenk (Hrsg.), Springer, 20, pp. 153-168.

[9] FutureID, *Survey and analysis of existing eID and credential systems,* Deliverable 32.1, 2013.

[10] Bundesamt für Sicherheit in derInformationstechnik, *Advanced Security Mechanism for Machine Readable Travel Documents - Extended Access Control (EAC), Password Authenticated Connection Establishment (PACE), and Restricted Identification (RI),* BSI-TR-03110, Version 2.10, 2012.

[11] T. Dierks und E. Rescorla, *The Transport Layer Security (TLS) Protocol Version 1.2,* 2008.

[12] AS Sertifitseerimiskeskus, *The Estonian ID Card and Digital Signature Concept - Principles and Solutions,* http://www.id.ee/public/The_Estonian_ID_Card_and_Digital_Signature_Concept.pdf, 2003.

[13] Population Register Centre, *FINeID Specification,* http://www.fineid.fi/default.aspx?id=590.

[14] J. Camenisch und E. Van Herreweghen, „Design and Implementation of the Idemix Anonymous Credential System," in *Proceedings of the 9th ACM Conference on Computer and Communications Security, CCS 2002, Washingtion, DC, USA, November 18-22, 2002*, 2002.

[15] IBM, *Identity Mixer,* http://idemix.wordpress.com/.

[16] Microsoft, *U-Prove,* http://www.microsoft.com/uprove.

[17] D. Bauer, D. M. Blough und D. Cash, „Minimal information disclosure with efficiently verifiable credentials," in *Proceedings of the 4th ACM workshop on Digital identity management (DIM '08)*, New York, 2008.

[18] E. Verheul, „Self-Blindable Credential Certificates from the Weil Pairing," in *ASIACRYPT*, Springer, LNCS 2248, 2001.

[19] D. Hühnlein, D. Petrautzki, J. Schmölz, T. Wich, M. Horsch, T. Wieland, J. Eichholz, A. Wiesmaier, J. Braun, F. Feldmann, S. Potzernheim, J. Schwenk, K. C., A. Kühne und H. Veit, *On the design and implementation of the Open eCard App,* GI SICHERHEIT 2012 Sicherheit - Schutz und Zuverlässigkeit, März 2012.

[20] M. Horsch, D. Hühnlein, C. Breitenstrom, T. Wieland, A. Wiesmaier, B. Biallowons, D. Petrautzki, S. Potzernheim, J. Schmölz, A. Wesner und T. Wich, *Die Open eCard App für mehr Transparenz, Vertrauen und Benutzerfreundlichkeit beim elektronischen Identitätsnachweis,* BSI-Kongress, Secumedia, 2013.

[21] S. Pötzsch, K. Borcea-Pfitzmann, M. Hansen, K. Liesebach, A. Pfitzmann und S. Steinbrecher, „Requirements for identity management from the perspective of multilateral interactions," in *Digital privacy*, Berlin, Heidelberg, Springer, 2011, pp. 609-649.

[22] M. Hansen und S. Thomsen, „Lebenslanger Datenschutz: Anforderungen an vertrauenswürdige Infrastrukturen," *Datenschutz und Datensicherheit,* pp. 283-288, 5 34 210.

[23] FutureID, *Privacy Requirements Analysis,* Deliverable D22.3, 2013.

[24] International Civil Aviation Organization (ICAO), Machine Readable Travel Documents, ICAO Doc 9303, Part 1-3, 2006 - 2008.

[25] European Committee for Standardization (CEN), Identification card systems - European Citizen Card, Part 1 - 4, CEN/TS 15480, 2008.

[26] Staatssekretariat für Wirtschaft (SECO), *SuisseID specification - Digital Certificates and Core Infrastructure Services,* Version 1.3, http://www.suisseid.ch, 2010.

[27] A-SIT, *The Austrian Citizen Card,* Version 1.2, http://www.buergerkarte.at/konzept/securitylayer/spezifikation/aktuell/, 2008.

[28] G. Hornung, M. Horsch und D. Hühnlein, „Mobile Authentisierung und Signatur mit dem neuen Personalausweis - Innovative technische und rechtliche Lösungsansätze," *Datenschutz und Datensicherheit (DuD),* Bd. 36, Nr. 3, pp. 189-194, 2012.

[29] J. Bender, M. Fischlin und D. Kügler, „Security Analysis of the PACE Key-Agreement Protocol," in *ISC 2009*, 2009.

[30] Ö. Dagdelen und M. Fischlin, „Security Analysis of the Extended Access Control Protocol for Machine Readable Travel Documents.," in *ISC 2010*, 2010.

[31] J. Bender, D. Kügler, M. Margraf und I. Naumann, „Das Sperrmanagement im neuen deutschen Personalausweis - Sperrmanagement ohne globale chipindividuelle Merkmale," *Datenschutz und Datensicherheit,* pp. 295-298, 5 34 2010.

[32] Bundesamt für Sicherheit in der Informationstechnik (BSI), *eCard-API-Framework,* Technical Guideline TR-03112, Part 1 - 7, Version 1.1.2, https://www.bsi.bund.de/ContentBSI/Publikationen/TechnischeRichtlinien/tr03112/index_htm.html.

[33] ISO/IEC, *Identification cards - Integrated circuit card programming interfaces,* ISO/IEC 24727, Part 1 - 5.

[34] T. Wich, D. Petrautzki, J. Schmölz, M. Horsch und D. Hühnlein, *An extensible platform for eID, signatures and more,* in Vorbereitung, 2013.

[35] Java.net, LiveConnect Support in the New Java™ Plug-In Technology, http://jdk6.java.net/plugin2/liveconnect/.

[36] RSA Laboratories, *PKCS #11 Base Functionality v2.30: Cryptoki,* 2009: http://www.cryptsoft.com/pkcs11doc/STANDARD/pkcs-11v2-30b-d5.doc.

[37] Microsoft, „Cryptographic Service Providers," [Online]. Available: http://msdn.microsoft.com/en-us/library/windows/desktop/aa380245%28v=vs.85%29.aspx.

[38] The Legion of the Bouncy Castle, „Bouncy Castle Crypto API,“ [Online]. Available: http://www.bouncycastle.org/java.html.

[39] A. Herrick, JSR 56: Java Network Launching Protocol and API, Maintainance Release 6, 2011.

[40] J. Camenisch, M. Dubovitskaya, A. Lehmann, G. Neven , C. Paquin und F.-S. Preiss, „Concepts and languages for privacy-preserving attribute-based authentication,“ in *IFIP IDMAN* , http://www.neven.org/papers/abc_credspec.html, 2013.

[41] J. Camenisch, I. Krontiris, A. Lehmann, G. Neven, C. Paquin und K. Rannenberg, *ABC4Trust Architecture for Developers,* https://abc4trust.eu/index.php/pub/149-h2-1, 2012.