Dokumentation

Security-Layer Applikationserhebung

Version 1.0, 24.03.2016

Tobias Kellner – tobias.kellner@egiz.gv.at
Arne Tauber – arne.tauber@egiz.gv.at

Zusammenfassung: Der Security-Layer ist eine proprietäre Schnittstelle, die 2001 für Zwecke der Bürgerkartenkommunikation (Personenbindung, Signatur, etc.) entwickelt wurde. In diesem Dokument wurde analysiert, welche Applikationen in Österreich in welchem Umfang die Security-Layer Schnittstelle nutzen, in welchem Umfang die Security-Layer-Schnittstelle erhalten bleiben soll bzw. wie mögliche Migrationsszenarien auf aktuelle Standards (OASIS SAML, DSS, ...) aussehen könnten.

E-Government Innovationszentrum
Inffeldgasse 16/a, A-8010 Graz
Tel. +43 316 873 5514
Fax. +43 316 873 5520
E-Mail post@egiz.gv.at
www.egiz.gv.at

Das E-Government Innovationszentrum ist eine gemeinsame Einrichtung des Bundeskanzleramtes und der TU-Graz

Inhaltsverzeichnis

1 Einleitung

Der Security-Layer ist eine proprietäre Schnittstelle, die 2001 für Zwecke der Bürgerkartenkommunikation (Personenbindung, Signatur, etc.) entwickelt wurde. In diesem Projekt soll analysiert werden, welche Applikationen in Österreich in welchem Umfang die Security-Layer Schnittstelle nutzen. Basierend auf dieser Analyse soll entschieden werden in welchem Umfang die Security-Layer-Schnittstelle erhalten bleiben soll bzw. wie mögliche Migrationsszenarien auf aktuelle Standards (OASIS SAML, DSS, ...) aussehen könnten.

In einem ersten Schritt wurde eine möglichst vollständige Liste jener derzeit in Verwendung befindlichen Anwendungen erstellt, welche die Security-Layer-Schnittstelle verwenden. Daraufhin wurde untersucht, soweit die jeweilige Anwendung öffentlich verfügbar ist, in welchem Umfang die Security-Layer-Spezifikation [SecL] in diesen Anwendungen umgesetzt ist.

Aus dieser Übersicht über den in Verwendung befindlichen Umfang der Security-Layer-Schnittstelle ergeben sich in weiterer Folge Szenarien, diese Schnittstelle durch entsprechende verfügbare Standard-Protokolle mittel- bzw. langfristig zu ersetzen, oder zumindest den Umfang der Spezifikation zu reduzieren. Dies muss auch mit weiteren Entwicklungen, speziell mit der neuen Architektur im eIDAS Kontext (Online-Personenbindung usw.), abgestimmt werden.

2 Security-Layer-Anwendungen Übersicht

2.1 Einleitung

Hier wird versucht, Security-Layer-Anwendungen grob in Kategorien einzuteilen. Die hierbei ermittelten Kategorien sind:

- Bürgerkarten-Umgebungen
- Anmeldung
  - Elektronische Zustellung (e-Delivery)
  - e-Banking
- Dokumentensignatur
  - Elektronische Formulare
- Dokumentenspeicher
- Beschaffung (e-Procurement)
- Rechnungslegung (e-Invoicing)
- Bürgerkarten-Aktivierung
- Sonstiges

Im Folgenden werden die einzelnen Kategorien kurz beschrieben:

2.2 Übersicht

2.2.1 Bürgerkarten-Umgebungen

Keine Anwendung im Sinne dieser Untersuchung ist die Bürgerkarten-Umgebung die Middleware, die auf Seite des Benutzers die Security-Layer-Schnittstelle umsetzt. Als solche muss sie einen minimalen Umfang der Security-Layer-Befehle unterstützen, und kann darüber hinaus weitere Befehle unterstützen.

2.2.2 Anmeldung

Bei diesen Anwendungen wird die Bürgerkarte verwendet, um den Benutzer zu identifizieren und zu authentifizieren. In den meisten Fällen wird hierfür [MOA-ID] verwendet, es gibt jedoch auch eigene Umsetzungen.

2.2.3 Elektronische Zustellung (e-Delivery)

Hierbei wird die Bürgerkarte lediglich zur Anmeldung benutzt (konkret wird MOA-ID verwendet), diese Anwendungen fallen also in die obige Kategorie. Der einzige Unterschied besteht darin, dass im Zuge der Anmeldung nicht der MOA-ID Standardtext, sondern eine Erweiterung davon, unterzeichnet wird, um den Empfang von Zustellstücken zu bestätigen.

2.2.4 e-Banking

Auch hier wird die Bürgerkarte zur Anmeldung verwendet, allerdings wird dafür nicht MOA-ID eingesetzt. Es kommt ausschließlich der Signaturpart zum Tragen, der Identifikationsteil mit Personenbindung entfällt beim e-Banking.

2.2.5 Dokumentensignatur

Hier wird die Bürgerkarte eingesetzt, um unterschiedliche Dokumente rechtsgültig zu unterschreiben. Meist handelt es sich hierbei um entweder XML- oder PDF-Dokumente, die nach [XAdES]- bzw.

[PAdES]-Standard unterschrieben werden. XML-Dokumente kommen zumeist im Umfeld von Bürgersignaturen wie MOA-ID und Formularlösungen zum Einsatz, PDF bei Amtssignaturen.

2.2.6 Elektronische Formulare

Hierbei werden Benutzerdaten aufgenommen und anschließend elektronisch signiert. Es handelt sich also um einen Sonderfall der vorigen Kategorie. Meist werden zu diesem Zweck die aufgenommenen Daten in einem XML-Dokument gesammelt und mit einer XML-Signatur versehen.

2.2.7 Dokumentenspeicher

Unter dieser Kategorie werden teils recht unterschiedliche Anwendungen zusammengefasst. Einerseits gibt es öffentlich verfügbare Cloud-Dokumentenspeicher wie e-Tresor oder Handy-Signatur, bei denen die Bürgerkarte einerseits zur Anmeldung (siehe 2.2.2) und andererseits zur Dokumentensignatur (siehe 2.2.5) eingesetzt wird.

Andererseits gibt es spezielle Anwendungen wie cyberDOC, das Urkundenarchiv des österreichischen Notariats, oder Archivium, das elektronische Dokumentenarchiv für Rechtsanwälte.

2.2.8 Beschaffung (e-Procurement)

e-Beschaffung (e-Procurement) ist die elektronische Beschaffung von Gütern und Dienstleistungen. Dabei können Aufträge öffentlich vergeben und Ausschreibungen verwaltet werden, beziehungsweise an Auschreibungen von potentiellen Auftragnehmern teilgenommen werden.

Zur rechtlichen Verbindlichkeit werden hierbei elektronische Signaturen eingesetzt.

2.2.9 Rechnungslegung (e-Invoicing)

Bei der elektronischen Rechnungslegung (e-Invoicing) können Rechnungen elektronisch erstellt werden, die die gleiche Rechtswirkung wie Papierrechnungen haben, ohne dass diese ausgedruckt werden müssten.

Auch hier kommen elektronische Signaturen zum Einsatz.

2.2.10 Bürgerkarten-Aktivierung

Zur Aktivierung einer Bürgerkarte ist es nötig, die Personenbindung [PersB] sowie die nötigen qualifizierten Zertifikate auf die Bürgerkarte aufzubringen bzw. zu aktivieren. Dafür gibt es besondere Security-Layer-Befehle.

3 Derzeitige Security-Layer-Verwendung

3.1 Einführung

Im Folgenden werden die aktuell in Verwendung befindlichen Security-Layer-Anwendungen daraufhin untersucht, in welchem Umfang sie die im Security-Layer spezifizierten Befehle verwenden. Dafür werden sie in die im vorigen Kapitel identifizierten Anwendungs-Kategorien eingeteilt.

3.2 Übersicht Security-Layer-Befehle

Befehl	Kurzbeschreibung
CreateCMSSignatureRequest	Signatur nach [CAdES] erstellen
CreateXMLSignatureRequest	Signatur nach XAdES erstellen
VerifyCMSSignatureRequest	Signatur nach CAdES prüfen
VerifyXMLSignatureRequest	Signatur nach XAdES prüfen
EncryptCMSRequest	Verschlüsselung als CAdES-Nachricht
EncryptXMLRequest	Verschlüsselung als XAdES-Nachricht
DecryptCMSRequest	Entschlüsselung einer CAdES-Nachricht
DecryptXMLRequest	Entschlüsselung einer XAdES-Nachricht
CreateHashRequest	Hashwert-Berechnung
VerifyHashRequest	Hashwert-Verifikation
InfoboxAvailableRequest	Abfrage verfügbarer Infoboxen
InfoboxCreateRequest	Anlegen einer Infobox
InfoboxDeleteRequest	Löschen einer Infobox
InfoboxReadRequest	Lesen von Infobox-Daten
InfoboxUpdateRequest	Verändern von Infobox-Daten
NullOperationRequest	Null-Operation
GetPropertiesRequest	Abfrage der Umgebungseigenschaften
GetStatusRequest	Abfrage des Tokenstatus

Hierbei ist zu beachten, dass PAdES-PDF-Signaturen auf CAdES-Signaturen beruhen.

3.3 Übersicht Security-Layer-Protokoll

Als Transportprotokoll werden im Security-Layer sowohl TCP/IP und SSL/TLS als auch HTTP und HTTPS spezifiziert, in der Praxis wird aber lediglich die HTTP(S)-Bindung verwendet.

Bei der HTTP(S)-Bindung sind einige spezielle Formular-Parameter definiert:

- `XMLRequest` – übermittelt den Security-Layer-Befehl
- `RedirectURL` – Leitet den Browser nach erfolgter Abarbeitung auf diese URL
- `DataURL` – Nach erfolgter Abarbeitung wird die Antwort an diese URL gesendet, von dort können auch weitere Security-Layer-Befehle übermittelt werden. Dies macht komplexere Abläufe möglich.
- `StylesheetURL` – Lädt eine XSL-Transformation von der angegebenen URL um die Befehlsantwort damit zu transformieren

Weiters gibt es noch einige Parameter, die gesondert ausgewertet werden:

- Weitergabe-Parameter – Formularelemente, deren Name auf einem Unterstrich endet, werden an die DataURL weitergereicht
- Weitergabe-Header – Formularelemente, deren Name auf zwei Unterstrichen endet, werden als HTTP-Header an die DataURL übermittelt.

Alle oben genannten Parameter werden so auch in der Praxis eingesetzt.

3.4 Übersicht Anwendungen

3.4.1 Bürgerkarten-Umgebungen

Derzeit verfügbare Bürgerkarten-Umgebungen:

- MOCCA (Online und Lokal)
- A-Trust a.sign client
- IT Solution trustDesk
- BDC hotSign

Laut Security-Layer-Spezifikation [1] müssen zumindest folgende Befehle umgesetzt werden:

- `CreateCMSSignatureRequest`
- `CreateXMLSignatureRequest`
- `InfoboxAvailableRequest`
- `InfoboxCreateRequest`
- `InfoboxDeleteRequest`
- `InfoboxReadRequest`
- `InfoboxUpdateRequest`
- `NullOperationRequest`
- `GetPropertiesRequest`
- `GetStatusRequest`

In der Praxis werden aber meist nicht alle dieser Befehle (vollständig) umgesetzt – siehe auch:
https://www.buergerkarte.at/konzept/securitylayer/spezifikation/20140114/tutorial/tutorial.html#uebersicht-bku-support

3.4.2 Anmeldung

Üblicherweise wird, um eine Anmeldung an einem Anwendungsportal zu starten, erst die Personenbindung ausgelesen um den Benutzer zu identifizieren, und dann in weiterer Folge ein Dokument ausgeliefert, welches durch den Benutzer unterschrieben wird und somit nach erfolgreicher Prüfung die Anmeldung autorisiert.

Bei allen hier untersuchten Anmeldungen kommt entweder direkt [MOA-ID] zum Einsatz, oder aber andere Software, welche allerdings nach genau demselben Prinzip vorgeht. Hierzu werden die Befehle `InfoboxReadRequest` zum Auslesen der Personenbindung, sowie `CreateXMLSignatureRequest` zum Erstellen der Signatur verwendet.

[1] https://www.buergerkarte.at/konzept/securitylayer/spezifikation/20140114/minimum/minimum.html

Der `InfoboxReadRequest` enthält dabei einen weiteren Parameter `IdentityLink-DomainIdentifier`, der den Bereich oder die Stammzahl enthält, um in der Personenbindung das (w)bPK zu bilden.

3.4.2.1 Beispiele für MOA-ID-Anmeldungen:

- FinanzOnline
- help.gv.at
- MOA-ID Demoanmeldung

3.4.2.2 Anmeldungen, bei denen eventuell andere Software zum Einsatz kommt:

- RTR e-Portal
- egov-service.at
- FabaSoft Cloud

3.4.3 Elektronische Zustellung (e-Delivery)

Diese Kategorie ist ein Sonderfall der vorigen, da die Bürgerkarte hier zur Anmeldung verwendet wird. Es gilt also dasselbe wie im vorigen Absatz.

Zusätzlich bestätigt der Bürger mit seiner XML-Signatur den Empfang elektronisch zugestellter Schriftstücke. Das MOA-ID Signatur-Text-Template wurde hier entsprechend angepasst, um bei der Anmeldung gleichzeitig einen Zustellnachweis zu erbringen.

3.4.3.1 Elektronische Zustellservices

- Postserver
- MeinBrief
- BRZ
- E-Versand

3.4.3.2 Weitere Zustellservices außerhalb des Zustellgesetzes

- BriefButler

3.4.4 e-Banking

Auch beim e-Banking kommt die Bürgerkarte zur Anmeldung zum Einsatz, allerdings nicht mittels MOA-ID oder ähnlicher Software. Bei diesen Anwendungen wird durchgehend nicht die Personenbindung zur Identifikation herangezogen, sondern das Signaturzertifikat des Benutzers.

Somit wird lediglich der `CreateXMLSignatureRequest`-Befehl eingesetzt. Mittels dieser Signatur erfolgen die Autorisierung sowie die Identifikation über das zur Signatur eingesetzte qualifizierte Zertifikat. Dafür muss einmalig das e-Banking-Konto mit dem

entsprechenden Zertifikat verknüpft werden, indem sich der Benutzer zuerst bei seinem e-Banking-Account mit anderen Mitteln anmeldet, und danach eine XML-Signatur durchführt oder die Zertifikatsdaten (bspw. CIN) out-of-band übermittelt.

3.4.4.1 e-Banking-Anbieter

- Raiffeisen
- ARZ (banking.co.at)
- BAWAG/PSK/easyBank

3.4.5 Dokumentensignatur

Hierbei handelt es sich um unterschiedliche Anwendungen mit dem Ziel, ein Dokument qualifiziert elektronisch zu unterzeichnen. Der Ablauf ist dabei stets derselbe:

Wenn nötig wird zuerst mittels `InfoboxReadRequest` die Personenbindung ausgelesen, um für die Signatur notwendige Informationen über den Signator zu ermitteln, und daraufhin mittels `CreateXMLSignatureRequest` bzw. `CreateCMSSignatureRequest` die XML- oder CMS- (PDF-)Signatur berechnet.

Die meisten Anwendungen verwenden zur PDF-Signatur die PDF-AS-Bibliothek.

PDF-AS Version 3 erzeugt Signaturen in einem speziellen österreichischen Format, dem PDF-AS-(2.0-)Format. Hierbei handelt es sich um eine PDF-Signatur, die auf einer speziellen XAdES- (XMLDSig-)Signatur basiert. Außerdem wird ein spezieller visueller Signaturblock spezifiziert, der auf das signierte Dokument aufgebracht wird. Da es sich hier um keinen weit verbreiteten Standard handelt, können diese Signaturen nur mit eigenen Tools überprüft werden, die diesen PDF-AS-Standard unterstützen. Des Weiteren gibt es die Besonderheit der textuellen Signatur, die auch von einem Papierausdruck (durch vollständige Eingabe des gedruckten Textes sowie eines Prüfwertes) verifiziert werden kann. Dies war aufgrund gesetzlicher Bedingungen vonnöten, die heute nicht mehr gegeben sind. Diese Version wird allerdings immer noch eingesetzt.

Die Aktuelle Version dieses Standards, PDF-AS Version 4, basiert auf dem PAdES-Standard. Hierbei gibt es die Besonderheit, dass aufgrund der Forderung des Gesetzgebers, die Anzeige der Signaturdaten während des Signaturvorganges ermöglichen zu müssen (§18 SigG), ein bestimmter Bereich der übermittelten Signaturdaten von der eigentlich Signatur ausgenommen werden muss (jener Bereich, in den in weiterer Folge die Signatur eingefügt wird). Würde dieser Bereich nicht mitübermittelt werden, wäre dies kein gültiges PDF, und könnte somit von der

Bürgerkartenumgebung ohne zusätzliche Informationen dem Benutzer nicht angezeigt werden. Dies ist insbesondere im Hinblick auf mögliche Nachfolgeprotokolle wichtig.

Zur Signaturprüfung gäbe es auch entsprechende Security-Layer-Befehle, allerdings ist keine Anwendung bekannt, die die Bürgerkartenumgebung dafür nützen würde – die bekannten Prüfdienste prüfen die Signatur serverseitig oder lokal ohne Verwendung des Security-Layer.

3.4.5.1 Beispiele für Signaturdienste:

- PDF-AS-Web
- PrimeSign
- PDF-Over
- A-Trust Signaturservice für die Handy-Signatur
- easyVersand
- SignOnline

3.4.6 Elektronische Formulare

Ein Spezialfall der Dokumentensignatur, da hier lediglich die gesammelten Formulardaten elektronisch signiert werden – es gilt also dasselbe wie im vorigen Absatz.

Die meisten Behörden setzen hier auf die AFORMSOLUTION- (AFS-)Software der Firma aforms2web, es gibt aber auch Eigenlösungen.

3.4.6.1 Beispiele für Formulardienste:

- AForms2Web AFS
- SeGoF

3.4.7 Dokumentenspeicher

Eine eher heterogene Kategorie; Grundsätzlich wird der Security-Layer hier hauptsächlich zur Anmeldung verwendet (siehe 3.4.2), aber auch zur Dokumentensignatur (siehe 3.4.5).

Bei öffentlichen und privaten Cloud-Dokumentenspeichern wie e-Tresor oder Handy-Signatur wird eine elektronische Dokumentensignatur für hochgeladene Dokumente angeboten.

Bei auf spezielle Berufsgruppen spezialisierten Diensten wie cyberDOC oder Archivium werden XML-Signaturen und Verschlüsselung eingesetzt, um Berechtigungen nachzuweisen und Zugriffe zu bestätigen bzw. Daten vertraulich zu behandeln.

3.4.7.1 Beispiele:

- cyberDOC
- Archivium
- e-Tresor
- Handy-Signatur

3.4.8 Beschaffung (e-Procurement) und Rechnungslegung (e-Invoicing)

Hierbei handelt es sich um kostenpflichtige Dienste, es war daher nur ein sehr eingeschränktes Testen möglich. Grundsätzlich kann davon ausgegangen werden, dass sich die hier verwendeten Security-Layer-Befehle in die oben bereits erwähnte Hauptkategorie Dokumentensignatur (siehe 3.4.5) handelt, um Beschaffungen oder Rechnungen elektronisch zu unterfertigen.

3.4.8.1 Beispiele für Beschaffungssoftware:

- vemap
- ANKÖ (vergabeportal.at)
- IT Solution trustDesk Procure

3.4.8.2 Beispiele für Rechnungslegungssoftware:

- BDC hotBill / hotInvoice

3.4.9 Bürgerkarten-Aktivierung

Hierbei handelt es sich um die einzige Anwendung, bei der Befehle über die bis jetzt verwendeten hinaus eingesetzt werden.

Im Besonderen sind dies `InfoboxCreateRequest`/`InfoboxUpdateRequest` und weitere nicht spezifizierte Befehle, um spezielle Befehle auf der Karte ausführen zu können (dafür wurden mittlerweile die `CardManagementRequest` sowie `CardChannel-Request`-Befehle spezifiziert, allerdings sind diese noch nicht im Einsatz).

Mittels dieser Befehle wird beispielsweise die Personenbindung aufgebracht, ein Zertifikat aufgebracht, oder die verschiedenen PINs werden aktiviert. Diese Befehle werden nicht von allen Bürgerkartenumgebungen unterstützt.

3.5 Zusammenfassung

Zusammenfassend lässt sich sagen, dass von den derzeitig verwendeten Anwendungen hauptsächlich folgende Security-Layer-Befehle verwendet werden:

- `InfoboxReadRequest`
- `CreateXMLSignatureRequest`
- `CreateCMSSignatureRequest`
- `NullOperationRequest`

3.5.1 NullOperationRequest

`NullOperationRequest` wurde oben nicht erwähnt, wird aber häufig verwendet, um die Verfügbarkeit einer Bürgerkartenumgebung abzufragen, oder in weiterer Folge mittels `DataURL` andere Security-Layer-Transaktionen anzustoßen. Ansonsten hat dieser Befehl keine Auswirkungen.

3.5.2 CreateXMLSignatureRequest

Der `CreateXMLSignatureRequest` bietet einige zusätzliche Parameter:

- `KeyboxIdentifier` – hiermit wird ausgewählt, welcher Schlüssel zur Signatur verwendet werden soll (in der Praxis `SecureSignatureKeypair` oder `CertifiedKeypair`)
- `DataObjectInfo` – Informationen zum zu unterschreibenden Datenobjekt:
    - Structure – `enveloping` oder `detached`, um die entsprechende Signaturform auszuwählen
    - Das Datenobjekt selbst kann dabei in mehreren Formen übergeben werden (Base64-kodiert, direkt als XML-Dokument, per Referenz)
    - `TransformsInfo` – XML-Transformationen die vor der Signatur (und der Anzeige der Signaturdaten) auf das Datenobjekt angewandt werden
    - `MimeType` – der MIME-Typ des Datenobjekts (text/plain, application/xhtml+xml, ...)
        - Für HTML-Daten spezifiziert der Security-Layer ein eigenes Anzeigeformat, ein Subset von XHTML/CSS, das „Standard-Anzeigeformat“ [SAF]
    - `Supplements` – Ergänzungsobjekte, die zur Transformation herangezogen werden können (z.B. Stylesheets)

3.5.3 CreateCMSSignatureRequest

Auch der `CreateCMSSignatureRequest` hat einige Parameter:

- `Structure` – `enveloping` oder `detached`, um die entsprechende Signaturform auszuwählen

- `KeyboxIdentifier` – hiermit wird ausgewählt, welcher Schlüssel zur Signatur verwendet werden soll (in der Praxis `SecureSignatureKeypair` oder `CertifiedKeypair`)
- `DataObject` – Das Datenobjekt selbst sowie zusätzliche Informationen:
    - `Content` – Das zu signierende Datenobjekt
    - `MetaInfo` – Signierte Meta-Informationen wie z.B. MIME-Typ
    - `ExcludedByteRange` – Hiermit wird, wie vorher erwähnt, ein spezieller Bereich der Signaturdaten angegeben, der von der Signatur ausgenommen wird, aber zur Anzeige des Signaturobjekts verwendet wird
    - `PAdESCompatibility` – Wählt PAdES-Kompatibilität für die resultierende CAdES-Signatur

3.5.4 InfoboxReadRequest

Hierbei kann eingeschränkt werden, dass der `InfoboxReadRequest` lediglich zum Auslesen der Personenbindung verwendet wird. Der Security-Layer definiert verschieden Arten von InfoBoxen (z.B. Assoziative Arrays), welche aber infolgedessen praktisch nicht verwendet werden. Die Personenbindung wird als einfache Binärdatei gespeichert.

Auch hier werden beim Auslesen einige Parameter mit übergeben – für die Personenbindung sind dies:

- `InfoboxIdentifier` – Identifizierung der gewünschten Box, hier `IdentityLink`
- `BoxSpecificParameters` – Spezifische Parameter für diese Infobox:

`IdentityLinkDomainIdentifier` – Enthält den Bereich der Anwendung oder die Stammzahl des Auftraggebers, um aus der Stammzahl der Personenbindung das bereichsspezifische Personenkennzeichen (

- [bPK]) zu berechnen.

3.5.5 Andere Befehle

Andere Befehle, wie zum Beispiel jene zur Signaturprüfung oder auch Ver- und Entschlüsselung, werden in den ermittelten Anwendungen überhaupt nicht eingesetzt.

Weitere Befehle zur Manipulation von Infoboxen und Signaturkarte werden lediglich zur Kartenaktivierung eingesetzt.

4 Migrations-Szenarien

4.1 Einführung

Aufgrund der in den vorangegangen Kapiteln herausgearbeiteten doch sehr eingeschränkten Nutzung der Security-Layer-Schnittstelle wäre es durchaus wünschenswert, auf entsprechende Standardprotokolle umzusteigen, um Kompatibilität mit anderen und modernen Anwendungen zu gewährleisten beziehungsweise Entwicklungsaufwand zu reduzieren.

4.2 Protokoll

Das Security-Layer-Protokoll an sich hat viele Eigenheiten und lässt sich nicht direkt durch einen offenen Standard ersetzen. Derzeit verfügbare Anwendungen müssen daher bei Umstieg auf neue Protokolle auf jeden Fall angepasst werden, da sich die bisherige Protokollabfolge nicht direkt auf offene Alternativen umlegen lässt.

Betrachten wir hierzu die einzelnen Anwendungsgebiete:

4.2.1 Dokumentensignatur

Die Befehle zur Erzeugung (und Verifikation) von Dokumentsignaturen könnten durch den Digital Signature Services- ([DSS])-Standard ersetzt werden, ein XML-basiertes Protokoll zu Erzeugung und Verifikation von XAdES- und CAdES-Signaturen.

Einige Elemente des Security-Layer-Protokolls können hier allerdings nicht eins zu eins übertragen werden:

Schlüssel-Auswahl: hierfür gibt es in DSS eine Entsprechung – das `KeySelector`-Element

4.2.1.1 XML-Signaturen

Transformationen: in DSS können zwar Transformationen angegeben werden, allerdings werden diese, im Gegensatz zum Security-Layer, nicht vom Signaturendpunkt angewandt, sondern müssen bereits zum Anfragezeitpunkt vom Anfragenden durchgeführt worden sein.

4.2.1.2 CMS-Signaturen

Ein Problem bietet hier die unter 2.2.5 erwähnte Sonderbehandlung von PDF-Dokumenten, für die der DSS-Standard keine Möglichkeit bietet. Hier müsste eine andere technische Lösung gefunden werden, um das Signaturdokument aus den Signaturdaten zu rekonstruieren.

Sollte diese gesetzliche Vorgabe allerdings im Zuge der Umstellung auf die eIDAS-Regulierung nicht mehr gegeben sein, wäre auch hier ein einfacher Umstieg möglich.

4.2.2 Anmeldung

Im Speziellen also Identifikation und Authentifikation.

Hier kommen speziell für diese Anwendung geeignete Identitätsprotokolle wie zum Beispiel [SAML] oder [OpenIDC]onnect in Betracht.

Soll der Prozess der Anmeldung möglichst an den bisherigen Prozessablauf bei Security-Layer-Anwendungen angelehnt sein, würde man zuerst den Identifikations-Schritt durchführen, und dann den Authentifikations-Schritt mittels einer Dokumentensignatur – wofür die im vorigen Abschnitt erwähnte DSS-Schnittstelle in Frage käme. Was hierbei allerdings fehlt, ist die Bindung dieser beiden Schritte aneinander, was im Falle des Security-Layer-Protokolls durch Verwendung des `DataURL`-Features gegeben war.

Im Falle einer Bürgerkartenumgebung mit einer einzelnen Identität ist dieses Problem nicht relevant, sobald aber mehrere Identitäten ins Spiel kommen, muss man eine Lösung finden, wie diese beiden Prozesse gekoppelt werden könnten.

Um diesen Prozess innerhalb eines „Transport"-Protokolles zu halten, könnte, wie dies auch beim STORK-Projekt eingesetzt wird, der DSS-Signatur-Request in ein SAML-RequestedAttribute verpackt werden. Allerdings müsste man dann noch immer eine Möglichkeit finden, diese Requests einander zuzuordnen. Dies wäre beispielsweise über das `InResponseTo`-Element der `SubjectConfirmationData` des `Subject`s möglich.

Bleibt also lediglich die Identifikation. Eine Besonderheit, auf die dabei Rücksicht genommen werden muss, ist die Bildung des bereichsspezifischen Personenkennzeichens. Dafür muss der Bereich beziehungsweise die Stammzahl des Anfordernden übermittelt werden.

Im Fall eines zentralen „eID-Knotens" (z.B. OPB) in Österreich könnte man eine Entkoppelung von Identifikation und Signatur bspw. mittels Single-Sign-On (SSO) realisieren. Die SSO-Session müsste allerdings bis zum ZDA durchgeschleust werden. Auch wäre mit einem zentralen Knoten die Realisierung eines „Security-Layer-Proxies" möglich, der als Kompatibilitäts- bzw. Transformationsschicht zwischen alten Security-Layer-Protokollen und neuen Standards fungiert, womit bestehende Applikationen mit minimalem Aufwand in einem Migrationsszenario auf neue Protokolle adaptiert werden könnten.

4.2.2.1 SAML

SAML 2.0 ist ein offener Standard, der ein XML-basiertes Format spezifiziert, mittels dem Identifikations- und Authentifikationsdaten übermittelt werden können, sowie ein Protokoll um diese Übermittlung durchzuführen.

Um den Bereich beziehungsweise die Stammzahl zu übermitteln, kann das `RequestedAttribute`-Feature [2] verwendet werden. Mit einem entsprechend angepassten einfachen SAML 2.0-Profil könnte man also die Personenbindungsabfrage ersetzen.

4.2.2.2 OpenID Connect

OpenID Connect ist ebenfalls ein offener Standard, und spezifiziert eine Identifizierungs-Schicht auf dem [OAuth] 2.0-Protokoll. Dabei kann ein Endbenutzer authentifiziert werden, und in weiterer Folge können Identitätsdaten dieses Benutzers abgefragt werden.

Der Authentifikations-Schritt kann in diesem Ablauf entfallen, da zuerst lediglich die Identifikations-Daten abgefragt werden (dies muss zuerst passieren, da diese Daten in die Signatur zur Authentisierung einfließen).

Der Bereich bzw. die Stammzahl könnten beispielsweise durch eigens formatierte claims[3] übermittelt werden.

4.3 Reduktion des Umfangs der Security-Layer-Spezifikation

Wenn man die derzeit verwendeten Eigenschaften der Security-Layer-Spezifikation betrachtet, wäre eine radikale Kürzung möglich, sofern man nicht gleich auf standardisierte Protokolle wechseln möchte. Einen Sonderfall bietet die Karten-Aktivierung, allerdings ist diese Momentan ohnehin außerhalb der Security-Layer-Spezifikation.

Somit könnte man den Security-Layer auf folgendes reduzieren:

- HTTP(S)-Bindung in nahezu unverändertem Umfang (`RedirectURL`, `DataURL`, ...)
- `NullOperationRequest`
- `CreateXML/CMSSignatureRequest` in nahezu unverändertem Umfang
- `InfoboxReadRequest` mit `IdentityLinkDomainIdentifier` um die Personenbindung auszulesen

---

[2] http://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf

[3] http://openid.net/specs/openid-connect-core-1_0.html#AdditionalClaims

4.4 Zusammenfassung

Das Ersetzen der einzelnen Komponenten, die derzeit im Security-Layer-Protokoll verwendet werden, durch entsprechende Standard-Protokolle ist durchaus möglich, allerdings mit Aufwand verbunden, wenn der Prozessfluss so wie bisher erhalten werden soll. Abhilfe würden hier zentrale Services (wie OPB) mit Hilfe eines Security-Layer-Proxies schaffen, die die Abwärtskompatibilität zu bestehenden Services herstellen und somit eine Migration hin zu neuen Protokollen erleichtern würden.

Eine Vereinfachung der bisher bestehenden Security-Layer-Spezifikation wäre durchaus denkbar, da nur ein Bruchteil der zur Verfügung stehenden Möglichkeiten genutzt wird, und würde somit einen Vorteil für die Implementierung neuer Security-Layer-Anwendungen bringen, da sich die Komplexität erheblich reduziert.

5 Anhang

5.1 Applikations-Liste

Hier wird für mehrfach verwendete Anwendungen nur jeweils eine beispielhaft aufgeführt.

5.1.1 Bürgerkarten-Umgebungen

- MOCCA
- a.sign Client
- BDC hotSign
- trustDesk

5.1.2 Anmeldung

- MOA-ID
- OAuth SSO
- Portalverbund
- Fabasoft Cloud
- RTR

5.1.3 Elektronische Zustellung (e-Delivery)

- postserver
- MeinBrief
- BRZ
- eVersand
- BriefButler

5.1.4 e-Banking

- Raiffeisen
- ARZ (banking.co.at)
- BAWAG

5.1.5 Dokumentensignatur

- PDF-AS(-Web)
- PrimeSign
- Handy-Signatur
- BriefButler
- easyVersand
- PDF-Over
- SignOnline
- a.sign PDF/PDFVerify/client

- BDC hotPDFSign/hotPDFVerify
- trustDesk Professional
- Handy-Signatur Windows App
- sPDF
- Signaturprüfung

5.1.6 Elektronische Formulare

- AFormSolution Formserver
- SeGoF
- ÖKOM
- HPA
- Buergerportal.at
- egov-service.at (Gemdat)
- BRZ Formularservice
- IT-Kommunal
- Tirol XGovForms

5.1.7 Dokumentenspeicher

- cyberDOC
- Archivium
- e-Tresor
- Handy-Signatur

5.1.8 Beschaffung (e-Procurement)

- vemap Beschaffungssoftware
- ANKÖ (vergabeportal.at)
- trustDesk Procure

5.1.9 Rechnungslegung (e-Invoicing)

- BDC hotBill/hotInvoice

5.1.10 Bürgerkarten-Aktivierung

- A-Trust Aktivierung

Dokumentenhistorie

Version	Datum	Autor(en)	Anmerkung
0.1	10.03.2016	Tobias Kellner	Aus Bericht übertragen
0.2	18.03.2016	Tobias Kellner	Erweiterung Kapitel 1-3
0.3	22.03.2016	Arne Tauber	Review
0.4	22.03.2016	Tobias Kellner	Migrations-Szenarien
1.0	24.03.2016	Arne Tauber Tobias Kellner	Editorielle Korrekturen & Finalisierung Final review

Referenzen

[SecL] https://www.buergerkarte.at/konzept/securitylayer/spezifikation/aktuell/

[MOA-ID] https://www.egiz.gv.at/en/schwerpunkte/13-moaspssid

[XAdES] European Telecommunications Standards Institute: ETSI TS 101 903: Electronic Signatures and Infrastructures (ESI); XML Advanced Electronic Signatures (XAdES) v1.4.2, Technische Spezifikation, Dezember 2010

[PAdES] European Telecommunications Standards Institute: ETSI TS 102 778: Electronic Signatures and Infrastructures (ESI); PDF Advanced Electronic Signature Profiles v1.2.1, Technische Spezifikation, Juli 2010

[CAdES] European Telecommunications Standards Institute: ETSI TS 101 733: Electronic Signatures and Infrastructures (ESI); CMS Advanced Electronic Signatures (CAdES) v2.2.1, Technische Spezifikation, April 2013

[PersB] http://www.buergerkarte.at/konzept/personenbindung/spezifikation/aktuell/

[SAF] https://www.buergerkarte.at/konzept/securitylayer/spezifikation/20140114/viewerformat/viewerformat.html

[bPK] https://www.stammzahlenregister.gv.at/site/5972/default.aspx

[SAML] http://saml.xml.org/saml-specifications

[OpenIDC] http://openid.net/specs/openid-connect-core-1_0.html

[OAuth] Internet Engineering Task Force (IETF): RFC 6749: The OAuth 2.0 Authorization Framework, Technische Spezifikation, Oktober 2012

[DSS] https://docs.oasis-open.org/dss/v1.0/oasis-dss-core-spec-v1.0-os.html