Anforderungen an die Benutzer-Schnittstelle zur *Bürgerkarten-Umgebung* der österreichischen Bürgerkarte		Konvention 1.2.0 Empfehlung
Kurzbeschreibung	Das vorliegende Dokument spezifiziert die Anforderungen an die Benutzerschnittstelle zur *Bürgerkarten-Umgebung* nach dem Modell Bürgerkarte.	
Autoren:	Arno Hollosi Gregor Karlinger Thomas Rössler Martin Centner et al.	Projektteam/Arbeitsgruppe AG Bürgerkarte
Datum:	14.5.2004	

Inhaltsverzeichnis

1. Allgemeines
- 1.1. Schlüsselwörter
- 1.2. Namenskonventionen

2. Benutzerinteraktion im Rahmen des Zugriffsschutzes

3. Benutzerinteraktion bei den einzelnen Befehlen
- 3.1. Signaturerstellung
- 3.2. Signaturprüfung
- 3.3. Verschlüsselung
- 3.4. Entschlüsselung
- 3.5. Hashwert-Berechnung
- 3.6. Hashwert-Verifikation
- 3.7. Infoboxen
- 3.8. Abfrage von Eigenschaften
- 3.9. Null-Operation

Glossar

Referenzen

A. Historie

1. Allgemeines

Bei der Ausführung der meisten der in Applikationsschnittstelle Security-Layer spezifizierten Befehle ist es notwendig, dass die *Bürgerkarten-Umgebung* über die *Benutzer-Schnittstelle* mit dem *Bürger* kommuniziert. Dieses Dokument legt die Mindestanforderungen an diese Kommunikation fest.

1.1. Schlüsselwörter

Dieses Dokument verwendet die Schlüsselwörter MUSS, DARF NICHT, ERFORDERLICH, SOLLTE, SOLLTE NICHT, EMPFOHLEN, DARF, und OPTIONAL zur Kategorisierung der Anforderungen. Diese Schlüsselwörter sind analog zu ihren englischsprachigen Entsprechungen MUST, MUST NOT, REQUIRED, SHOULD, SHOULD NOT, RECOMMENDED, MAY, und OPTIONAL zu handhaben, deren Interpretation in [Keywords] festgelegt ist.

1.2. Namenskonventionen

Zur besseren Lesbarkeit wurde in diesem Dokument auf geschlechtsneutrale Formulierungen verzichtet. Die verwendeten Formulierungen richten sich jedoch ausdrücklich an beide Geschlechter.

Folgende Namenraum-Präfixe werden in dieser Spezifikation zur Kennzeichnung der Namenräume von XML-Elementen verwendet:

Präfix	Namenraum	Erläuterung
`sl`	`http://www.buergerkarte.at/namespaces/securitylayer/1.2#`	Elemente dieser Spezifikation

2. Benutzerinteraktion im Rahmen des Zugriffsschutzes

In der Regel wird die erste Kommunikation mit dem *Bürger* über die Benutzer-Schnittstelle im Rahmen der Prüfung erfolgen, ob ein Befehl von der *Bürgerkarten-Umgebung* ausgeführt werden soll oder nicht (vergleiche Abschnitt 3.1, „Regeln“ in Die österreichische Bürgerkarte - Zugriffsschutz). Je nach dem, wie die Regeln für die Ausführung eines bestimmten Befehls gesetzt sind, mündet die Regelprüfung in eine von vier Interaktionsarten. Die nachfolgende Auflistung legt fest, wie diese Interaktionsarten über die Benutzer-Schnittstelle abzubilden sind.

`none`

Es keine Interaktion mit dem *Bürger* notwendig.

`info`

Die *Bürgerkarten-Umgebung* MUSS die Befehlsausführung protokollieren. Folgende Informationen müssen zumindest protokolliert werden: Authentisierungsklasse, Identifikationsbegriff, Befehlsname, sowie allfällige Parameter des Befehls im Sinne von Abschnitt 3.1, „Regeln“ in Die österreichische Bürgerkarte - Zugriffsschutz. Das so geführte Protokoll MUSS dem *Bürger* über die Benutzer-Schnittstelle leicht zugänglich sein und in einfach zu verstehender Form dargestellt werden.

`confirm`

Die *Bürgerkarten-Umgebung* MUSS vor der Befehlsausführung die Erlaubnis des *Bürgers* einholen. Damit der *Bürger* diese Entscheidung fällen kann, MUSS ihm die *Bürgerkarten-Umgebung* zumindest folgende Informationen präsentieren: Authentisierungsklasse, Identifikationsbegriff, Befehlsname, sowie allfällige Parameter des Befehls im Sinne von Abschnitt 3.1, „Regeln“ in Die österreichische Bürgerkarte - Zugriffsschutz. Zusätzlich MUSS die Befehlsausführung wie für die Interaktionsart `info` beschrieben protokolliert werden.

`confirmWithSecret`

Analog zur Interaktionsart `confirm` MUSS die *Bürgerkarten-Umgebung* vor der Befehlsausführung die Erlaubnis des *Bürgers* einholen. Zusätzlich MUSS der *Bürger* im Falle einer positiven Entscheidung diese durch Übermittlung eines Kennworts an die *Bürgerkarten-Umgebung* dokumentieren. Die Übermittlung des Kennworts an eine serverbasierte *Bürgerkarten-Umgebung* MUSS in einer verschlüsselten Verbindung erfolgen. Zusätzlich MUSS die Befehlsausführung wie

für die Interaktionsart `info` beschrieben protokolliert werden.

3. Benutzerinteraktion bei den einzelnen Befehlen

Neben der oben erläuterten Benutzerinteraktionen im Rahmen des Zugriffsschutzes gibt es eine Reihe von Befehlen aus Applikationsschnittstelle Security-Layer, für die eine spezielle Art der Benutzerinteraktion ihm Rahmen der Befehlsabarbeitung durch die *Bürgerkarten-Umgebung* notwendig ist. Die nachfolgenden Ausführungen behandeln diese getrennt nach den einzelnen Befehlen.

3.1. Signaturerstellung

Vor der Erstellung der Signatur MUSS die *Bürgerkarten-Umgebung* dem *Bürger* die Möglichkeit bieten, das zu signierende Dokument bzw. die zu signierenden Dokumente (abhängig vom verwendeten Signaturformat) anzuzeigen.

Wenn eine *sichere elektronische Signatur* erzeugt werden soll, DARF die *Bürgerkarten-Umgebung* dem *Bürger* die Auslösung der elektronischen Signatur NICHT möglich machen, wenn die zu signierenden Dokumente mittels der Anzeigekomponente der *Bürgerkarten-Umgebung* nicht dargestellt werden können.

Wenn hingegen keine sichere elektronische Signatur erzeugt werden soll, DARF dem *Bürger* die Auslösung auch dann ermöglicht werden, wenn die zu signierenden Dokumente nicht mittels der Anzeigekomponente der *Bürgerkarten-Umgebung* dargestellt werden können. In einem solchen Fall SOLLTE die *Bürgerkarten-Umgebung* dem *Bürger* jedoch zumindest die Möglichkeit bieten, das zu signierende Dokument bzw. die zu signierenden Dokumente mit Hilfe einer externen Anzeigemöglichkeit darzustellen. Beispielsweise könnte die *Bürgerkarten-Umgebung* die Möglichkeit bieten, ein zu signierendes Dokument abzuspeichern, um es dann mit einer externen Software zu laden und anzuzeigen. Ebenfalls vorstellbar wäre es, dass die *Bürgerkarten-Umgebung* direkt eine externe Software aufruft, um ein zu signierendes Dokument darzustellen. Im letzten Fall MUSS die *Bürgerkarten-Umgebung* dem *Bürger* jedoch deutlich vermitteln, dass eine externe Software zur Anzeige verwendet wird.

Neben dem zu signierenden Dokument bzw. den zu signierenden Dokumenten MUSS die *Bürgerkarten-Umgebung* dem *Bürger* auch den Zeitpunkt der Signaturerstellung anzeigen, da diese Zeit als Signaturattribut mit den eigentlich zu signierenden Dokumenten mitsigniert wird (vergleiche Applikationsschnittstelle Security-Layer, Abschnitte 2.1.2 sowie 2.2.2). Sinnvollerweise wird die *Bürgerkarten-Umgebung* als diesen Zeitpunkt die Zeit des Aufrufs des Signaturerstellungsbefehls verwenden.

Vor dem Auslösen der Signaturfunktion und der damit gegebenenfalls verbundenen Eingabe einer Signatur-PIN MUSS die *Bürgerkarten-Umgebung* den *Bürger* auf deutliche Weise davon in Kenntnis setzen, mit welchem Schlüssel die Signatur erstellt werden soll. Eine dafür geeignete Weise ist die Darstellung des zum Schlüssel zugehörigen Zertifikats.

Im Rahmen des Auslösens der Signaturfunktion MUSS die *Bürgerkarten-Umgebung* dem *Bürger* die Möglichkeit bieten, sowohl die Signatur als auch die signierten Dokumente lokal zu speichern.

3.2. Signaturprüfung

Nach der erfolgten Signaturprüfung MUSS die *Bürgerkarten-Umgebung* dem *Bürger* eine übersichtliche Zusammenfassung der Prüfungsergebnisse darstellen. Jedenfalls dargestellt werden MUSS das Ergebnis der kryptographischen Prüfung (Hashwerte, Signaturwert). Konnte die Zertifikatsprüfung grundsätzlich durchgeführt werden (d. h. konnte ein zum verwendeten Schlüssel ausgestelltes Signatorzertifikat festgestellt werden), MUSS auch das Ergebnis dieser Prüfung (Gültigkeit des Signatorzertifikats, Vertrauenswürdigkeit des Signatorzertifikats) dargestellt werden. Weiters MUSS die *Bürgerkarten-Umgebung* dann dem *Bürger* auch alle wesentlichen Merkmale des Signatorzertifikats anzeigen. Als wesentlich sind jedenfalls der Name des Signators (*Subject DN*), der Name des ausstellenden Zertifizierungsdiensteanbieters (*Issuer DN*), die Seriennummer des Zertifikats sowie die allenfalls vorhandenen Zertifikatserweiterungen zur Kennzeichnung eines qualifizierten Zertifikats bzw. zur Kennzeichnung der Verwaltungseigenschaft einzustufen.

Ist in der zu prüfenden Signatur ein mitsigniertes Attribut enthalten, dass den vom Signator behaupteten Signaturerstellungszeitpunkt beinhaltet (vergleiche Applikationsschnittstelle Security-Layer, Abschnitte 2.1.2 sowie 2.2.2), MUSS die *Bürgerkarten-Umgebung* diesen Zeitpunkt ebenfalls dem *Bürger* darstellen und klar als die vom Signator behauptete Zeit kennzeichnen.

Die *Bürgerkarten-Umgebung* MUSS dem *Bürger*, wenn zumindest die kryptographische Prüfung erfolgreich durchgeführt werden konnte, die Möglichkeit bieten, sich die signierten Dokumente anzeigen

zu lassen. Ist eine Darstellung der signierten Dokumente mittels der Anzeigekomponente der *Bürgerkarten-Umgebung* nicht möglich, MUSS die *Bürgerkarten-Umgebung* dem *Bürger* eine Möglichkeit zur Verfügung stellen, die signierten Daten mit einer externen Anzeigemöglichkeit darzustellen. Beispielsweise könnte die *Bürgerkarten-Umgebung* die Möglichkeit bieten, ein signiertes Dokument abzuspeichern, um es dann mit einer externen Software zu laden und anzuzeigen. Ebenfalls vorstellbar wäre es, dass die *Bürgerkarten-Umgebung* direkt eine externe Software aufruft, um ein signiertes Dokument darzustellen. Im letzten Fall MUSS die *Bürgerkarten-Umgebung* dem *Bürger* jedoch deutlich vermitteln, dass eine externe Software zur Anzeige verwendet wird.

Schließlich MUSS die *Bürgerkarten-Umgebung* dem *Bürger* im Rahmen der Signaturprüfung die Möglichkeit bieten, sowohl die Signatur als auch die signierten Dokumente lokal zu speichern.

3.3. Verschlüsselung

Die *Bürgerkarten-Umgebung* MUSS dem *Bürger* die Möglichkeit bieten, alle zu verschlüsselnden Dokumente anzuzeigen, bevor sie die Befehlsantwort an die *Applikation* sendet. Ist eine Anzeige mittels der internen Anzeigekomponente nicht möglich, MUSS die *Bürgerkarten-Umgebung* dem *Bürger* eine Möglichkeit zur Verfügung stellen, die zu verschlüsselnden Dokumente mit einer externen Anzeigemöglichkeit darzustellen. Beispielsweise könnte die *Bürgerkarten-Umgebung* die Möglichkeit bieten, ein zu verschlüsselndes Dokument abzuspeichern, um es dann mit einer externen Software zu laden und anzuzeigen. Ebenfalls vorstellbar wäre es, dass die *Bürgerkarten-Umgebung* direkt eine externe Software aufruft, um ein zu verschlüsselndes Dokument darzustellen.

Vor dem Auslösen der Verschlüsselungsfunktion MUSS die *Bürgerkarten-Umgebung* den *Bürger* auf deutliche Weise davon in Kenntnis setzen, wer der Empfänger der verschlüsselten Daten sein wird. Eine dafür geeignete Weise ist die Darstellung des zum verwendeten Verschlüsselungs-Schlüssel zugehörigen Zertifikats.

3.4. Entschlüsselung

Im Falle eine lokalen *Bürgerkarten-Umgebung* kann es vorkommen, dass ein *Bürger* mehrere Krypto-Token zur Verwahrung von Entschlüsselungsschlüsseln besitzt. Für einen solchen Fall SOLLTE die *Bürgerkarten-Umgebung* dem *Bürger* einen Hinweis für die Auswahl des passenden Entschlüsselungsschlüssels bieten, sofern sie selbst Informationen darüber (z.B. durch das in den Verschlüsselungsdaten angegebene Zertifikat zum Verschlüsselungsschlüssel) kennt.

Ist für das Auslösen der Entschlüsselungsfunktion die Eingabe einer Verschlüsselungs-PIN notwendig, MUSS die *Bürgerkarten-Umgebung* den *Bürger* auf deutliche Weise davon in Kenntnis setzen, welcher Schlüssel für die Entschlüsselung verwendet wird. Eine dafür geeignete Weise ist die Darstellung des zugehörigen Zertifikats.

Bevor die *Bürgerkarten-Umgebung* die Befehlsantwort sendet, MUSS sie dem *Bürger* die Möglichkeit bieten, sich alle entschlüsselten Dokumente anzeigen zu lassen. Ist eine Anzeige mittels der internen Anzeigekomponente nicht möglich, MUSS die *Bürgerkarten-Umgebung* dem *Bürger* eine Möglichkeit zur Verfügung stellen, die entschlüsselten Dokumente mit einer externen Anzeigemöglichkeit darzustellen. Beispielsweise könnte die *Bürgerkarten-Umgebung* die Möglichkeit bieten, ein entschlüsseltes Dokument abzuspeichern, um es dann mit einer externen Software zu laden und anzuzeigen. Ebenfalls vorstellbar wäre es, dass die *Bürgerkarten-Umgebung* direkt eine externe Software aufruft, um ein entschlüsseltes Dokument darzustellen.

Weiters MUSS die *Bürgerkarten-Umgebung*, bevor sie die Befehlsantwort sendet, dem *Bürger* die Möglichkeit bieten, alle entschlüsselten Dokumente lokal zu speichern.

3.5. Hashwert-Berechnung

Bevor sie die Befehlsantwort sendet, MUSS die *Bürgerkarten-Umgebung* dem *Bürger* das Ergebnis der Hashwert-Berechnung darstellen. Diese Darstellung MUSS je berechnetem Hashwert zumindest folgende Merkmale umfassen:

- Den zur Hashwert-Berechnung verwendeten Algorithmus;
- den berechneten Hashwert;
- den im Request angegebenen Friendly Name für das zu hashende Dokument;
- eine Möglichkeit zur Anzeige sowie zur lokalen Speicherung des zu hashenden Dokuments.

Ist eine Anzeige mittels der internen Anzeigekomponente nicht möglich, MUSS die *Bürgerkarten-Umgebung* dem *Bürger* eine Möglichkeit zur Verfügung stellen, ein zu hashendes Dokument mit einer externen Anzeigemöglichkeit darzustellen. Beispielsweise könnte die *Bürgerkarten-Umgebung* die Möglichkeit bieten, ein zu hashendes Dokument abzuspeichern, um es dann mit einer externen Software zu laden und anzuzeigen. Ebenfalls vorstellbar wäre es, dass die *Bürgerkarten-Umgebung* direkt eine externe Software aufruft, um ein zu hashendes Dokument darzustellen.

Weiters MUSS die *Bürgerkarten-Umgebung* dem *Bürger* die Möglichkeit bieten, die oben erwähnten Merkmale für alle während einer Sitzung der *Bürgerkarten-Umgebung* ausgeführten Hashwert-Berechnungen leicht zugänglich über die Benutzerschnittstelle abzurufen. Bei einem solchen späteren Abrufen MUSS die *Bürgerkarten-Umgebung* dem *Bürger* neben den erwähnten Merkmalen auch den Zeitpunkt (Datum und Uhrzeit) darstellen, zu dem die Hashwert-Berechnung durchgeführt wurde.

3.6. Hashwert-Verifikation

Bevor sie die Befehlsantwort sendet, MUSS die *Bürgerkarten-Umgebung* dem *Bürger* das Ergebnis der Hashwert-Verifikation darstellen. Diese Darstellung MUSS je verifiziertem Hashwert zumindest folgende Merkmale umfassen:

- Den zur Hashwert-Verifikation verwendeten Algorithmus;
- den Referenz-Hashwert aus der Befehlsanfrage;
- den berechneten Kontroll-Hashwert;
- das Ergebnis der Hashwert-Verifikation;
- den im Request angegebenen Friendly Name für das zu hashende Dokument;
- eine Möglichkeit zur Anzeige sowie zur lokalen Speicherung des zu hashenden Dokuments.

Ist eine Anzeige mittels der internen Anzeigekomponente nicht möglich, MUSS die *Bürgerkarten-Umgebung* dem *Bürger* eine Möglichkeit zur Verfügung stellen, ein zu hashendes Dokument mit einer externen Anzeigemöglichkeit darzustellen. Beispielsweise könnte die *Bürgerkarten-Umgebung* die Möglichkeit bieten, ein zu hashendes Dokument abzuspeichern, um es dann mit einer externen Software zu laden und anzuzeigen. Ebenfalls vorstellbar wäre es, dass die *Bürgerkarten-Umgebung* direkt eine externe Software aufruft, um ein zu hashendes Dokument darzustellen.

Weiters MUSS die *Bürgerkarten-Umgebung* dem *Bürger* die Möglichkeit bieten, die oben erwähnten Merkmale für alle während einer Sitzung der *Bürgerkarten-Umgebung* ausgeführten Hashwert-Verifikationen leicht zugänglich über die Benutzerschnittstelle abzurufen. Bei einem solchen späteren Abrufen MUSS die *Bürgerkarten-Umgebung* dem *Bürger* neben den erwähnten Merkmalen auch den Zeitpunkt (Datum und Uhrzeit) darstellen, zu dem die Hashwert-Verifikation durchgeführt wurde.

3.7. Infoboxen

3.7.1. Abfrage verfügbarer Infoboxen

Bevor sie die Befehlsantwort zusammenstellt, MUSS die *Bürgerkarten-Umgebung* dem *Bürger* die Möglichkeit bieten, aus der Menge der tatsächlich in der *Bürgerkarten-Umgebung* vorhandenen Infoboxen jene auszuwählen, die in die Befehlsantwort aufgenommen werden sollen. Der *Bürger* kann auf diesem Weg die Sichtbarkeit seiner Infoboxen einschränken.

3.7.2. Anlegen

Bevor die *Bürgerkarten-Umgebung* die Infobox anlegt und die Befehlsantwort an die *Applikation* sendet, MUSS sie dem *Bürger* die Parameter der anzulegenden Infobox aus der Befehlsanfrage anzeigen und ihm die Möglichkeit geben, einzelne Parameter zu verändern, sowie das Anlegen der Infobox abzulehnen.

Die Anzeige der Parameter aus der Befehlsanfrage MUSS folgende Elemente umfassen:

- Name der Infobox (`sl:InfoboxIdentifer`);
- Typ der Infobox (`sl:InfoboxType`);
- Informationen zur *Applikation*, welche die Infobox anlegen möchte (`sl:Creator`);
- Informationen zum Zweck der Infobox (`sl:Purpose`);

- Vorschlag für die Berechtigungen zum Lesezugriff (`sl:ReadAccessAuthorization`);
- Vorschlag für die Berechtigungen zum Änderungszugriff (`sl:UpdateAccessAuthorization`);
- Vorschlag für die Bestätigung eines Lesezugriffs (`sl:ReadUserConfirmation`);
- Vorschlag für die Bestätigung eines Änderungszugriffs (`sl:UpdateUserConfirmation`).

Der Name sowie der Typ der Infobox DÜRFEN vom *Bürger* NICHT verändert werden können. Die Informationen zur anlegenden *Applikation* sowie zum Zweck der Infobox MÜSSEN vom *Bürger* verändert werden können.

Ist das Attribut `UserMayChange` in der Befehlsanfrage auf den Wert `true` gesetzt, MUSS der *Bürger* den entsprechenden Vorschlag für eine Berechtigung bzw. Bestätigung verändern können. Ist der Wert des Attributs `UserMayChange` hingegen auf `false` gesetzt, DARF der Vorschlag vom *Bürger* NICHT verändert werden können.

Fehlen in der Befehlsanfrage einzelnen Vorschläge für Berechtigungen und Bestätigungen, MUSS die *Bürgerkarten-Umgebung* dem *Bürger* entsprechende Vorschläge machen, die der *Bürger* verändern können MUSS.

Ist der *Bürger* mit dem Anlegen der Infobox grundsätzlich, oder mit bestimmten unveränderbaren Vorschlägen für Berechtigungen und Bestätigungen nicht einverstanden, MUSS er die Möglichkeit haben, das Anlegen der Infobox abzulehnen.

3.7.3. Löschen

Für diesen Befehl ist keine befehlsspezifische Benutzerinteraktion vorgeschrieben. Die Regelungen zur Interaktion im Rahmen des Zugriffsschutzes bleiben davon unberührt.

3.7.4. Lesen

Wird in der Befehlsanfrage zum Lesen von Schlüsseln oder zum Lesen von Schlüsseln und Werten aus einer Infobox mit dem Typ *Assoziatives Array* das Attribut `UserMakesUnique` auf den Wert `true` gesetzt, und führt der in der Befehlsanfrage angegebene Suchbegriff zu mehr als einem passenden Schlüssel, MUSS die *Bürgerkarten-Umgebung* den *Bürger* aus der Menge der passenden Schlüssel genau einen auswählen lassen. In der Befehlsantwort MUSS dann genau dieser eine Schlüssel bzw. dieser eine Schlüssel zusammen mit dem zugeordneten Wert an die *Applikation* gesendet werden. Führt die oben erläuterte Befehlsanfrage zu keinem oder genau einem Treffer, SOLLTE die beschriebene Benutzerinteraktion NICHT stattfinden.

3.7.5. Verändern

Für diesen Befehl ist keine befehlsspezifische Benutzerinteraktion vorgeschrieben. Die Regelungen zur Interaktion im Rahmen des Zugriffsschutzes bleiben davon unberührt.

3.8. Abfrage von Eigenschaften

3.8.1. Umgebung

Für diesen Befehl ist keine befehlsspezifische Benutzerinteraktion vorgeschrieben. Die Regelungen zur Interaktion im Rahmen des Zugriffsschutzes bleiben davon unberührt.

3.8.2. Token

Sind in der Befehlsanfrage die beiden Elemente `sl:TokenStatus` und `sl:MaxDelay` vorhanden, und entspricht der aktuelle Status des Bürgerkarten-Tokens nicht dem in der Befehlsanfrage angegebenen, so MUSS die *Bürgerkarten-Umgebung* den *Bürger* über die Benutzerschnittstelle dazu auffordern, den gewünschten Status herzustellen. Läßt der *Bürger* die in MaxDelay vorgegebene Zeitspanne verstreichen, ohne den gewünschten Zustand herzustellen, MUSS die *Bürgerkarten-Umgebung* der *Applikation* in der Befehlsantwort den unverändert gebliebenen Status zurückmelden.

3.9. Null-Operation

Für diesen Befehl ist keine befehlsspezifische Benutzerinteraktion vorgeschrieben. Die Regelungen zur

Interaktion im Rahmen des Zugriffsschutzes bleiben davon unberührt.

Glossar

Glossar

Applikation

Jenes Programm, das Anfragen an die *Bürgerkarten-Umgebung* über den *Security-Layer* richtet und die entsprechenden Antworten entgegennimmt und auswertet.

Benutzer-Schnittstelle

Jene Schnittstelle, über die der *Bürger* mit der *Bürgerkarten-Umgebung* kommuniziert. Über diese Schnittstelle wird einerseits die Benutzerinteraktion abgewickelt, die gegebenenfalls zur Abwicklung eines Befehls des *Security-Layers* notwendig ist (z.B. die Anzeige eines zu signierenden Dokuments beim Befehl zur Erzeugung einer XML-Signatur); andererseits kann der *Bürger* über diese Schnittstelle seine *Bürgerkarten-Umgebung* nach seinen persönlichen Bedürfnissen konfigurieren (z.B. kann er Einstellungen zum Zugriffsschutz auf seine Infoboxen verändern). Die Vorgaben an die *Benutzer-Schnittstelle* sind in Minimale Umsetzung des Security-Layers geregelt.

Bürger

Jene Person, die die Funktionen der *Bürgerkarten-Umgebung* für die sichere Abwicklung von E-Government oder E-Commerce verwenden möchte. Die Ansteuerung der *Bürgerkarten-Umgebung* erfolgt in der Regel nicht durch den *Bürger* selbst, sondern durch die *Applikation*, welche die E-Government oder E-Commerce Anwendung repräsentiert.

Bürgerkarte

Laut [E-GovG], §10 ZI 10 ist die *Bürgerkarte* „die unabhängig von der Umsetzung auf unterschiedlichen technischen Komponenten gebildete logische Einheit, die eine elektronische Signatur mit einer Personenbindung (§ 4 Abs. 2) und den zugehörigen Sicherheitsdaten und -funktionen sowie mit allenfalls vorhandenen Vollmachtsdaten verbindet“. Im Sinne der in den Spezifikationen zur österreichischen Bürgerkarte gebrauchten Terminologie ist die *Bürgerkarten-Umgebung* die Implementierung der logischen Einheit *Bürgerkarte*.

Bürgerkarten-Umgebung

Jenes Programm bzw. jener Dienst, der die Funktionalität der *Bürgerkarte* zur Verfügung stellt. Grundsätzlich vorstellbar ist die Ausführung als Programm, das lokal am Rechner des *Bürgers* läuft (*lokale Bürgerkarten-Umgebung*), oder als serverbasierter Dienst, der über das Internet angesprochen wird (*serverbasierte Bürgerkarten-Umgebung*). Die Interaktion mit diesem Programm bzw. Dienst wird über zwei Schnittstellen abgewickelt: Über die *Benutzer-Schnittstelle* sowie über den *Security-Layer*.

Hash-Eingangsdaten

Jene Daten, die für die Berechnung des Hash-Wertes für eine `dsig:Reference` verwendet werden. Sind für die `dsig:Reference` Transformationen angegeben, entsprechen diese Daten dem Ergebnis der letzten Transformation. Sind keine Transformationen spezifiziert, gleichen die Hash-Eingangsdaten den *Referenz-Eingangsdaten*.

Impliziter Transformationsparameter

Siehe Abschnitt 2.2.2.2, „Implizite Transformationsparameter“ in Die österreichische Bürgerkarte - Applikationsschnittstelle Security-Layer

Referenz-Eingangsdaten

Jene Daten, die sich aus der Auflösung der im Attribut `URI` der `dsig:Reference` angegebenen URI ergeben. Sind für die `dsig:Reference` Transformationen angegeben, werden diese Daten als Eingangsdaten zur Berechnung der ersten Transformation verwendet. Sind keine Transformationen spezifiziert, gleichen die Referenz-Eingangsdaten den *Hash-Eingangsdaten*.

Security-Layer

Jene Schnittstelle, über die die *Applikation* mit der *Bürgerkarten-Umgebung* kommuniziert. Das genaue Protokoll, das über diese Schnittstelle gesprochen werden kann, wird in Applikationsschnittstelle Security-Layer spezifiziert. Die möglichen Bindungen dieses Protokolls an Transportschichten wie HTTP oder TCP wird in Transportprotokolle Security-Layer geregelt.

Signaturmanifest

Siehe Abschnitt 2.2.2.2, „Implizite Transformationsparameter“ in Die österreichische Bürgerkarte - Applikationsschnittstelle Security-Layer .

Referenzen

[CMS] BHously, R.: *RFC 3369: Cryptographic Message Syntax (CMS)* , IETF Request For Comment, August 2002

[CMS-AES] chaad, J.: *RFC 3565: Use of the Advanced Encryption Standard (AES) Encryption Algorithm in Cryptographic Message Syntax (CMS)* . IETF Request For Comment, Juli 2003.

[CMS-Alg] Hously, R.: *RFC 3370: Cryptographic Message Syntax (CMS) Algorithms* . IETF Request For Comment, August 2002.

[CMS-RSAES-OAEP] Hously, R.: *RFC 3560: Use of the RSAES-OAEP Key Transport Algorithm in the Cryptographic Message Syntax (CMS)* . IETF Request For Comment, Juli 2003.

[CSS 2] Bert Bos, Håkon Wium Lie, Chris Lilley und Ian Jacobs: *Cascading Style Sheets, level 2* . W3C Recommendation, Mai 1998.

[EC14N] Boyer, John, Eastlake, Donald und Reagle, Joseph: *Exclusive XML Canonicalization. W3C Recommendation, Juli 2002* .

[ECDSA-CMS] Blake-Wilson, S., Brown, D., Lampert, D.: *RFC 3278: Use of Elliptic Curve Cryptography (ECC) Algorithms in Cryptographic Message Syntax (CMS)* . IETF Request For Comment, April 2002.

[ECDSA-XML] Blake-Wilson, S., Karlinger, G. und Wang, Y.: *ECDSA with XML-Signature Syntax* . Internet-Draft, Jänner 2004.

[E-GovG] *BGBl. I Nr. 10/2004.*

[ESS-S/MIME] Hoffman, P.: *RFC 2634: Enhanced Security Services for S/MIME* , IETF Request For Comment, Juni 1999

[ETSICMS] European Telecommunications Standards Institute: *ETSI TS 101733: Electronic Signature Formats, v1.5.1* , Technical Specification, Dezember 2003

[ETSIQCert] European Telecommunications Standards Institute: *ETSI TS 101 862: Qualified certificate profile, v1.2.1* , Technical Specification, Juni 2001

[ETSIXML] European Telecommunications Standards Institute: *ETSI TS 101903: XML Advanced Electronic Signatures (XAdES), v1.2.2* , Technical Specification, April 2004

[GIF] *Graphics Interchange Format, Version 89a* . CompuServe Incorporated, Juli 1990.

[HTML4] Dave Ragget, Arnaud Le Hors und Ian Jacobs: *HTML 4.01 Specification* . W3C Recommendation, Dezember 1999.

[HTTP1.1] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leech und T. Berners-Lee: *Hypertext Transfer Protocol -- HTTP/1.1*. IETF Request For Comment, Juni 1999.

[HTTPS] E. Rescorla *HTTP over TLS*. IETF Request For Comment, Mai 2000

[ISO-8859-1] *ISO/IEC 8859-1*:1998: Information technology -- 8-bit single-byte coded graphic character sets -- Part 1: Latin alphabet No. 1.

[ISO-8859-10] *ISO/IEC 8859-10*:1998: Information technology -- 8-bit single-byte coded graphic character sets -- Part 10: Latin alphabet No. 6.

[ISO-8859-15] *ISO/IEC 8859-15*:1999: Information technology -- 8-bit single-byte coded graphic character sets -- Part 15: Latin alphabet No. 9.

[ISO-8859-2] *ISO/IEC 8859-2*:1999: Information technology -- 8-bit single-byte coded graphic character sets -- Part 2: Latin alphabet No. 2.

[ISO-8859-3] *ISO/IEC 8859-3*:1999: Information technology -- 8-bit single-byte coded graphic character sets -- Part 3: Latin alphabet No. 3.

[ISO-8859-9] *ISO/IEC 8859-9*:1999: Information technology -- 8-bit single-byte coded graphic character sets -- Part 9: Latin alphabet No. 5.

[JPEG] Eric Hamilton: *JPEG File Interchange Format, Version 1.02* . C-Cube Microsystems, September 1992.

[KEYWORDS] Bradner, S.: *RFC 2119: Key words for use in RFCs to Indicate Requirement Levels* , IETF Request For Comment, März 1997

[MIME] Freed, N. und Borenstein, N.: *RFC 2046: Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types* , IETF Request For Comment, November 1996

[PersBin] Hollosi, Arno und Karlinger, Gregor: *XML-Definition der Personenbindung* . Konvention zum E-Government Austria erarbeitet von der Stabsstelle IKT-Strategie des Bundes, Technik und Standards. Öffentlicher Entwurf, Version 1.2.2, 14. Februar 2005.

[PersonData] Naber, Larissa: *PersonData Struktur - XML Spezifikation* . Konvention zum E-Government Austria erarbeitet von der Arbeitsgruppe Kommunikationsarchitekturen. Öffentlicher Entwurf, Version 2.0.0, 14. Oktober 2004.

[PKCS#12] RSA Laboratories: *PKCS#12 v1.0: Personal Information Exchange Syntax* , Juni 1999.

[port-numbers] Internet Assigned Numbers Authority: *Port Numbers*

[QCert] Santesson, S. und Nystrom M.: *RFC 3739: Internet X.509 Public Key Infrastructure: Qualified Certificates Profile* , IETF Request For Comment, März 2004

[SigG] *BGBl I Nr. 190/1999* idF BGBl I Nr. 152/2001.

[SigV] *BGBl II Nr. 30/2000* idF BGBl II Nr. 527/2004.

[Stammzahl] Hollosi, Arno und Hörbe, Rainer: *Bildung von Stammzahl und bereichsspezifischem Personenkennzeichen (bPK)* . Konvention zum E-Government Austria erarbeitet von der Stabsstelle IKT-Strategie des Bundes, Technik und Standards sowie vom Bundesminsterium für Inneres. Öffentlicher Entwurf, Version 1.0, 2. Februar 2004.

[TLS] T. Dierks und C. Allen: *The TLS Protocol Version 1.0* . IETF Request For Comment, Januar 1999.

[Unicode] The Unicode Consortium. *The Unicode Standard, Version 4.0.0* , defined by: The Unicode Standard, Version 4.0 (Boston, MA, Addison-Wesley, 2003. ISBN 0-321-18578-1).

[URI] Berners-Lee, T. , Fielding, R. und Masinter, L.: *RFC 2396: Uniform Resource Identifiers (URI): Generic Syntax* , IETF Request For Comment, August 1998

[VerwEig] Hollosi, Arno: *X.509 Zertifikatserweiterungen für die Verwaltung* . Konvention zum E-Government Austria erarbeitet von der Stabsstelle IKT-Strategie des Bundes, Technik und Standards. Öffentlicher Entwurf, Version 1.0.3, 21. Februar 2005.

[X509] Polk, W., Ford, W., Solo, D.: *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile* . IETF Request For Comment, April 2002.

[XHTML 1.1] Murray Altheim, Frank Boumphrey, Sam Dooley, Shane McCarron, Sebastian Schnitzenbaumer und Ted Wugofski: *Modularization of XHTML* . W3C Recommendation, April 2001.

[XHTML MOD] Daniel Austin, Subramanian Peruvemba, Shane McCarron, Masayasu Ishikawa: *Modularization of XHTML in XML Schema* . W3C Working Draft, Oktober 2003.

[XML] Bray, Tim, Paoli, Jean, Sperberg-McQueen, C.M. und Maler, Eve: *Extensible Markup Language (XML) 1.0 (Second Edition)* , W3C Recommendation, Oktober 2000.

[XMLDecTF] Hughes, Merlin, Imamura, Takeshi und Maruyama, Hiroshi: *Decryption Transform for XML Signature* . W3C Recommendation, Dezember 2002.

[XMLDSIG] Eastlake, Donald, Reagle, Joseph und Solo, David: *XML-Signature Syntax and Processing* , W3C Recommendation, Februar 2002

[XMLDSIG-URI] Eastlake, Donald: *RFC 4051: Additional XML Security Uniform Resource Identifiers (URIs)* , IETF Request For Comments, April 2005

[XMLEnc] Eastlake, Donald und Reagle, Joseph: *XML Encryption Syntax and Processing* , W3C Recommendation, Dezember 2002

[XML-Schema] Thompson, Henry S., Beech, David, Maloney, Murray und Mendelson, Noah: *XML Schema Part 1: Structures* , W3C Recommendation, Mai 2001

[XMLTYPE] Murata, M., St.Laurent, S., und Kohn, D.: *RFC 3023: XML Media Types* , IETF Request For Comment, Jänner 2001.

[XPath] Clark, James und DeRose, Steven: *XML Path Language* , W3C Recommendation, November

1999

[XPF2] Boyer, John, Hughes, Merlin und Reagle, Joseph: *XML-Signature XPath Filter 2.0* . W3C Candidate Recommendation, Juli 2002.

[XPointer] Grosso, Paul, Maler, Eve, Marsh, Jonathan und Walsh, Norman: *XPointer Framework* . W3C Recommendation, März 2003.

[XSS-FAQ] Cgisecurity.com: *The Cross Site Scripting FAQ* .

A. Historie

Datum	Version	Änderungen
14. 05. 2004	1.2.0	• Erstellt.