EGIZ E-Government Innovationszentrum

www.egiz.gv.at
E-Mail: post@egiz.gv.at
Telefon: ++43 (316) 873 5514
Fax: ++43 (316) 873 5520
Inffeldgasse 16a / 8010 Graz / Austria

PDF Signatur/Amtssignatur

Spezifikation

Version 2.1.1

Inhalt

Dokument-Historie

Datum	Version	Autor / Organisation	Änderungen
25.08.2006	0.9.0	Wilfried Lackner (IICM) Wolfgang Prinz (IICM)	Dokument erstellt.
4.10.2006	1.0.0	Thomas Rössler (EGIZ)	Dokument formatiert, tw. korrigiert.
31.07.2007	1.1.0	Arian Mavriqi (IICM) Ernad Besirevic (IICM)	Dokument der Version 1.1.0 angepasst
16.01.2008	2.0.0 DR1	Thomas Rössler (EGIZ)	Neufassung.
24.01.2008	2.0.0	Thomas Rössler (EGIZ)	Gegengelesen Thomas Knall, Fertigstellung.
29.04.2008	2.0.1	Thomas Knall (EGIZ)	Die Einbettung der textuellen Signatur v1.1.0 MUSS mittels inkrementellem Update erfolgen.
03.02.2009	2.1	Thomas Knall (EGIZ)	Ergänzung hinsichtlich eines neuen Signaturparameters für die Open-Source BKU "MOCCA", Korrekturen, Überarbeitung und Vereinheitlichung des Layouts, Lesbarere Formatierung der Listings
18.03.2009	2.1.1	Thomas Knall (EGIZ)	In beiden Abschnitten für Binärsignatur "Einbettung der Signatur in das PDF-Dokument": "[...] Der gesamte Signaturblock [...]" geändert zu "Der gesamte *sichtbare* Signaturblock". Hinsichtlich EGIZ-Dictionary wurde folgender Satz (2x) eingefügt: "Das EGIZ-Dictionary DARF noch weitere Elemente enthalten. Diese können dazu verwendet werden die Signatur mit zusätzlicher Information auszustatten."

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 RFC 2119 festgelegt ist.

Begriffserklärungen

Binäre Signatur:

Eine binäre Signatur signiert das gesamte Dokument in binärer Repräsentation.

Detached Signatures:

Bei dem Detached-Modus wird kein Datenobjekt in die Signaturstruktur eingebunden d.h. die Signatur referenziert das Datenobjekt. Das Datenobjekt wird über diese Referenz erhalten. Vgl. dazu Definition aus [4]:

> *Signature, Detached*
>
> *The signature is over content external to the Signature element, and can be identified via a URI or transform. Consequently, the signature is "detached" from the content it signs. This definition typically applies to separate data objects, but it also includes the instance where the Signature and data object reside within the same XML document but are sibling elements.*

Enveloping Signatures:

Das Datenobjekt wird in die Signaturstruktur eingebunden. Vgl. dazu Definition aus [4]:

> *Signature, Enveloping*
>
> *The signature is over content found within an Object element of the signature itself. The Object (or its content) is identified via a Reference (via a URI fragment identifier or transform).*

Mehrfachsignatur:

Ein Dokument ist mehrfach hintereinander signiert. Sofern nicht anders im Dokument erwähnt, wird im Rahmen dieser Spezifikation von einer seriellen Mehrfachsignatur ausgegangen. Das heißt, durch mehrfaches Anwenden der hier spezifizierten Signaturprozesse können mehrere Signaturen hintereinander auf ein Dokument aufgebracht werden. Eine nachfolgende Signatur enthält dabei alle zuvor auf das Dokument aufgebrachten Signaturen und behandelt diese wie gewöhnliche Elemente eines Dokuments. Bei der Verifikation von Mehrfachsignatur muss dies entsprechend berücksichtigt werden.

Portable Document Format (PDF):

Das Portable Document Format, kurz PDF, hat sich in der Vergangenheit als das Standard Format zum Transport und zur Speicherung digitaler Inhalte etabliert. Für die Langzeitspeicherung digitaler Inhalte in Archiven wurde der auf PDF 1.4 aufsetzende PDF/A Standard entwickelt. Immer mehr Behörden nutzen diese Standards um Dokumente digital abzulegen. Daher ist es von vitalem Interesse diese digitalen PDF Dokumente auch sicher signieren und verifizieren zu können.

Prüfvorgang:

Der Vorgang der Prüfung (Verifikation) eines signierten PDF Dokuments wird als Prüfvorgang bezeichnet.

Signaturblock:

Der Signaturblock ist jener Teil des sichtbaren PDF Dokuments, welcher anzeigt, dass ein Dokument signiert ist.

Signaturvorgang:

Der Vorgang des Erstellens einer Signatur für ein gegebenes PDF Dokument wird als Signaturvorgang bezeichnet.

Textuelle Signatur:

Eine textuelle Signatur signiert den textuellen Inhalt eines Dokuments.

Signaturattribute:

Werte einer elektronischen Signatur, die diese charakterisieren, und die im Zuge des Signaturvorgangs erstellt werden. Zum Beispiel: Signaturwert, Signaturzeitpunkt, Signatureigenschaften (Signature-Properties), etc.

1 Einleitung

Dokumente im PDF-Format sind breit in Verwendung und im Online-Verkehr besonders etabliert - mehr als 200 Millionen PDF-Dokumente im Internet zeugen davon. Um auch im E-Government auf dieses beliebte Dokumentenformat zurückgreifen zu können - bspw. zur Kommunikation von der Behörde hin zum Bürger - müssen PDF-Dokumente auch mit einer elektronischen Signatur versehen werden können. Gerade im Falle von offiziellen Dokumenten der Behörde - wie etwa Bescheiden - werden durch das E-Government Gesetz (E-GovG) der auf die Dokumente aufzubringenden (Amts-)Signatur besondere Formvorschriften auferlegt.

Im Rahmen dieser Spezifikation wird ein Verfahren definiert, mit dem PDF-Dokumente mit einer elektronischen Signatur versehen werden können, die bei Bedarf selbst vom Papierausdruck rekonstruiert und verifiziert werden kann. Zum Aufbringen der Signatur können dabei verschiedene Signaturerstellungskomponenten verwendet werden, wie bspw. die Bürgerkarte oder aber ein serverseitiges Signaturmodul (MOA-SS).

Es werden zwei Methoden definiert, wie PDF-Dokumente signiert werden können:

- textuelle PDF-Signatur
- binäre PDF-Signatur

Die textuelle Signatur extrahiert nur den Text aus einem gegebenen PDF-Dokument, ignoriert jedoch Bilder und andere nicht textuelle Elemente, und signiert diesen Text in einer normalisierten Weise. So ist gewährleistet, dass textuell signierte PDF-Dokumente jederzeit auch auf Basis eines Papierausdruckes rekonstruiert und letztlich auch deren Signatur geprüft werden kann. Dieses Verfahren eignet sich besonders zur sicheren Signatur rein textueller PDF-Dokumente ohne grafische oder bildhafte Komponenten.

Ergänzend dazu wird die binäre PDF-Signatur spezifiziert, die zwar das gesamte PDF-Dokument mit allen darin enthaltenen Elementen signiert, deren Signatur aber letztlich nicht mehr von einem Ausdruck rekonstruiert werden kann.

Die Definition beider Signaturtypen, sowie deren theoretische Grundlagen, werden in diesem Dokument definiert.

***Anmerkung**: Der im Rahmen dieser Spezifikation definierte Typ von PDF-Signaturen wird im Verlauf dieses Dokuments mit „PDF-Amtssignatur“ bezeichnet. Es wird ausdrücklich darauf hingewiesen, dass trotz dieser Bezeichnung das Anwendungsfeld nicht auf Behördenapplikationen beschränkt zu sehen ist, sondern dass diese Signaturen selbstverständlich auch in allen "amtsfremden" Anwendungsbereichen bzw. der Privatwirtschaft analog einsetzbar sind.*

2 Charakteristika von PDF-Amtssignaturen

PDF-Amtssignaturen werden durch zwei Identifikationsbegriffe charakterisiert:

- Signaturmethode (Methode)
- Signaturparameter (Parameter)

Die Signaturmethode legt fest, auf welche Art und Weise das zu signierende Dokument im Zuge des Signaturprozesses behandelt wurde. Die Signaturmethode nimmt also Bezug auf die Vorbehandlung des PDF-Dokuments und auf jenen Prozess, der letztlich zu einem von einer Signaturerstellungskomponente zu signierenden Datenstrom führt.

Der Signaturparameter gibt an, welche Rahmenbedingungen im Zuge der Signaturerstellung vorlagen und unter welchen Bedingungen die Signatur technisch erzeugt wurde. Der Signaturparameter berücksichtigt demnach Spezifika der Signaturerstellungsprozedur sowie der herangezogenen Signaturerstellungskomponenten.

Diese zwei Charakteristika der PDF-Amtssignatur ergeben sich aufgrund der beiden ineinandergreifenden Prozesse, nämlich der Aufbereitung des zu signierenden Dokuments und des Signaturprozesses.

Die nachfolgenden Abschnitte spezifizieren diese beiden Charakteristika von PDF-Amtssignaturen im Detail.

2.1 Methode

Die Signaturmethode – im weiteren Verlauf nur als Methode bezeichnet – legt fest, auf welche Art und Weise das zu signierende Dokument im Zuge des Signaturprozesses behandelt wurde. Die Methode nimmt also Bezug auf die Vorbehandlung des PDF-Dokuments und auf jenen Prozess, der letztlich zum von einer Signaturerstellungskomponente zu signierenden Datenstrom führt.

Eine Methode ist ein Verarbeitungsprozess, der als Eingangsobjekt (Input-Datenstrom) das zu signierende PDF-Dokument heranzieht und am Ende den durch die Signaturerstellungskomponente weiterzuverarbeitenden und zu signierenden Datenstrom erzeugt.

Für jede Methode MUSS der jeweilige Verarbeitungsprozess spezifiziert und veröffentlicht werden. Als Input-Datenstrom für den Verarbeitungsprozess MUSS das zu signierende PDF-Dokument in Form eines binären Datenstroms herangezogen werden. Das Ergebnis des Verarbeitungsprozesses MUSS ein binärer Datenstrom sein, dessen MIME-Type ebenfalls durch die Spezifikation der Methode festgelegt werden MUSS.

Jeder spezifizierten Methode MUSS eine eindeutige Kennzeichnung vergeben werden, die in der optischen Repräsentation der PDF-Amtssignatur sichtbar dargestellt werden MUSS. Zur Kennzeichnung von Methoden MUSS folgende Notation ([9]) herangezogen werden:

```
<MethodeID>  ::= "urn:" <NID> ":" <NSS>
<NID>        ::= "pdfsigfilter"
<NSS>        ::= <VENDOR> ":" <METHODE> ":" <VERSION>
<VENDOR>     ::= "bka.gv.at" | 1*<URN chars>
<METHODE>    ::= "text" | "binaer" | 1*<URN chars>
<VERSION>    ::= "v" 1*<number> "." 1*<number> "." 1*<number>
<URN chars>  ::= siehe <URN chars> in RFC 2141
<number>     ::= siehe <number> in RFC 2141
```

<MethodeID>	Die Kennzeichnung der Methode.
<NID>	Der Namespace Identifier der URN. Dieser wird konstant mit „pdfsigfilter“ festgelegt.
<NSS>	Der Informationsblock der URN.
<VENDOR>	Eindeutiger Identifikationsbegriff jener Organisation, die den durch die vorliegende Kennzeichnung repräsentierte Methode festgelegt und spezifiziert hat.
<METHODE>	Identifikationsbegriff der Methode bzw. Methodenklasse, welche im Zuge der Signaturerstellung zur Anwendung gebracht wurde. Im Zuge der vorliegenden Kernspezifikation wurden zwei Methoden eingeführt: - textuelle Signatur ("text") - binäre Signatur ("binaer") Weiter Methoden sind möglich.
<VERSION>	Die exakte Version der angewandten Methode.

Beispiele:

```
urn:pdfsigfilter:bka.gv.at:binaer:v1.0.0
urn:pdfsigfilter:bka.gv.at:text:v1.1.0
```

Der Abschnitt 4 dieser Spezifikation definiert Signaturmethoden im Detail.

2.2 Parameter

Der Signaturparameter – im weiteren Verlauf nur als Parameter bezeichnet – gibt an, welche Rahmenbedingungen im Zuge der Signaturerstellung vorlagen und unter welchen Bedingungen die Signatur technisch erzeugt wurde. Der Parameter berücksichtigt demnach Spezifika der Signaturerstellungsprozedur sowie der herangezogenen Signaturerstellungskomponenten.

Grundsätzlich sind Parameter für die verschiedenen Signaturerstellungskomponenten notwendig. Bezieht sich allerdings die Spezifikation einer Signaturmethode bereits auf eine konkrete Standardsignaturkomponente, so KANN für Signaturen dieser Standardsignatur-komponenten die Angabe von Signaturparametern entfallen. In anderen Fällen SOLLEN Spezifika der Signaturerstellungskomponenten in Form eines Signaturparameters in der optischen Repräsentation der PDF-Amtssignatur lesbar enthalten sein.

Zur Angabe des Signaturparameters MUSS die folgende Struktur ([9]) angewendet werden:

```
<PARAMETER_ID> ::= <SIGDEV_PROF> "@" [<PARAM_L1>] ["@" [<PARAM_L2>]]
   <SIGDEV_PROF> ::= 1*<CHAR> | ""
   <PARAM_L1>    :: = 1*<CHAR>
   <PARAM_L2>    :: = 1*<CHAR>
   <CHAR>        :: = siehe <CHAR> in Abschnitt 6.1 in RFC 2234
```

<PARAMETER_ID>	Der Signaturparameter.
<SIGDEV_PROF>	Kennzeichnung des Signaturerstellungskomponente oder des konkret angewandten Signaturparameter-Profils (optional, falls notwendig).
<PARAM_L1>	Konkrete Parameter Teil 1 (Ebene 1). Diese werden konkret für eine Signaturerstellungskomponente festgelegt.
<PARAM_L2>	Konkrete Parameter Teil 2 (Ebene 2). Diese werden konkret für eine Signaturerstellungskomponente festgelegt.

Beispiele:

```
etsi-bka-1.0@1155648477-25748375@3902-22389-0-16078-16384
moass-1.0@1234ABC@
etsi-moc-1.0@b62b3b59
foo@@1234abcd
```

Parameter MÜSSEN in für den Leser sichtbaren Feldern im Signaturblock stehen.

Der Abschnitt 5 dieser Spezifikation definiert sogenannte Signaturparameter-Profile im Detail.

2.3 Anforderungen an PDF Dokumente

Ein zu signierendes PDF Dokument DARF NICHT verschlüsselt sein und SOLL sich an die Vorgaben der PDF-Spezifikation Version 1.4. [2] halten.

Sämtliche durch die hier spezifizierten von der Signaturmethodik bedingten Änderungen am und im zu signierenden PDF-Dokument MÜSSEN ebenfalls dem PDF Standard Version 1.4 entsprechen. Daher MUSS ein signiertes PDF-Dokument ebenfalls PDF-Standard Version 1.4 konform sein.

Im Zuge des Signaturvorganges DARF das zu signierende PDF-Dokument durch den Signaturvorgang NICHT verändert werden. Dies könnte bereits vorhandene binäre Signaturen zerstören oder beschädigen. Deshalb MÜSSEN Signaturen mittels eines PDF Incremental Update (siehe PDF Reference 1.4 [2], Kapitel 3.4.5) dem zu signierenden PDF-Dokument angefügt werden, sofern dies im Rahmen der Definitionen einer Signaturmethode nicht anderslautend festgelegt wird. Davon abweichende bzw. darüberhinausgehende Vorgaben KÖNNEN in der Definition von Signaturmethoden getroffen werden.

3 Repräsentation einer PDF-Amtssignatur

Wesentliche Eigenschaft einer PDF-Amtssignatur ist die visuelle Repräsentation der Signaturdaten im PDF-Dokument selbst. Anhand dieser soll nicht nur der Umstand eines signierten Dokumentes eindeutig erkennbar sein, sondern in besonderen Fällen sogar die Verifikation der Signatur auf Basis eines Papierausdrucks ermöglicht werden (Rekonstruktion). Demnach müssen alle zur Rekonstruktion einer elektronischen Signatur erforderlichen Werte visuell in der Repräsentation vorkommen.

Das Layout der Darstellung von Signaturblöcken in PDF-Dokumenten soll auch ein möglichst einheitliches sein, um einerseits einen konsistenten Auftritt gegenüber den BürgerInnen zu erreichen, und andererseits um die technische Rekonstruktion von Amtssignaturen zu erleichtern. Abbildung 1 gibt einen Vorschlag für das Layout einer rekonstruierbaren Amtssignatur.

Signaturwert	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	
	**Unterzeichner**	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
	Datum/Zeit-UTC	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
	Aussteller-Zertifikat	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
	Serien-Nr.	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
	Methode	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
	Parameter	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Prüfhinweis	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	

Abbildung 1: Muster einer visuellen Ausprägung der Amtssignatur

185 Das Layout bzw. die Anordnung der einzelnen Felder sowie die Bezeichnung der Felder KANN
186 frei gewählt werden. Semantisch MÜSSEN die folgenden Vorgaben für eine visuellen
187 Repräsentation eingehalten werden:

#	Feld-Bezeichnung (Signaturattribut)	M/K/S	Beschreibung
1	**Signaturwert**	MUSS	Signaturwert; ist erforderlich.
2	**Unterzeichner**	KANN	Name des Unterzeichners; ist ein optionales Feld und kann zur Verdeutlichung des Unterzeichners verwendet werden.
3	**Datum/Zeit-UTC**	MUSS	Datum und Zeitpunkt der Signatur (im UTC-Format); ist erforderlich.
4	**Aussteller-Zertifikat**	MUSS	Angaben zum Aussteller des Signaturzertifikates, zumindest dessen Namen und Herkunftsland; ist erforderlich.
5	**Serien-Nr.**	MUSS	Seriennummer des Signaturzertifikates; ist erforderlich.
6	**Methode**	MUSS	Element zur näheren Kennzeichnung des verwendeten Signaturverfahrens (Signaturmethode). Dieses Element kann verwendet werden, um bspw. den angewandten Signaturstandard zu identifizieren.
7	**Parameter**	KANN	Optionales Element zur Formulierung von für das/den angewandte Signaturverfahren/-standard notwendigen näheren Bestimmungsparametern. Dieses Feld ist sozusagen eine detailliertere und zusätzliche Möglichkeit, weitere Signaturparameter anzuführen; diese sind vom angewandten Signaturstandard bzw. von der verwendeten Signaturtechnologie abhängig.
8	**Prüfhinweis**	SOLL	Ein einfach verständlicher Hinweis für BürgerInnen, wie man die gegenständliche Amtssignatur verifizieren kann. Hierin kann bspw. ein Verweis auf ein Prüfservice im Internet beschrieben werden. Dieses Feld soll bei einem Signaturblock immer verwendet werden, um den BürgerInnen eine Unterstützung bei der Prüfung zu bieten. Hierin soll jedenfalls ein Hinweis stehen, ob und wie die gegenständliche Signatur auf Basis eines Papierausdruckes rekonstruiert, rückgeführt und geprüft werden kann.
9	[Bildmarke] keine textuelle Bezeichnung	SOLL	Die Bildmarke ist das optische und bildhafte Pendant zum Rundsiegel; ist erforderlich.

188 Konkrete weitere Feld-Bezeichner KÖNNEN bei Bedarf hinzugenommen werden. Es wird
189 EMPFOHLEN, sich bei der Wahl der Feld-Bezeichner sowie für das Layout der Repräsentation
190 insgesamt an Mustervorlagen anzulehnen. Eine entsprechende Empfehlung für den
191 Verwaltungsbereich ist mit [8] veröffentlicht.

4 Definierte Signaturmethoden

Die vorliegende Spezifikation definiert eine Reihe von Signaturmethoden, die wie folgt in Implementierungen unterstützt werden MÜSSEN:

Signaturmethode	Status	In Implementierungen zu unterstützen bei	
		Verifikation	Signatur
urn:pdfsigfilter:bka.gv.at:text:v1.0.0	DEPRECATED	EMPFOHLEN	NICHT EMPFOHLEN
urn:pdfsigfilter:bka.gv.at:text:v1.1.0	EMPFOHLEN	MUSS	MUSS
urn:pdfsigfilter:bka.gv.at:binaer:v1.0.0	DEPRECATED	EMPFOHLEN	NICHT EMPFOHLEN
urn:pdfsigfilter:bka.gv.at:binaer:v1.1.0	EMPFOHLEN	MUSS	MUSS

Eine spezifikationskonforme Umsetzung MUSS die in der obigen Tabelle definierten Signaturmethoden, gemäß den definierten Prioritäten, implementieren.

Jede Implementierung MUSS für die damit erzeugbaren Signaturmethoden sowohl die Signaturerstellung als auch die Signaturverifikation realisieren.

Nachfolgend werden die einzelnen Signaturmethoden im Detail definiert. Die Definition der Signaturmethoden und der darin enthalten Verarbeitungsschritte erfolgt aus Sicht des Signaturprozesses. Im Zuge einer Signaturverifikation ist daher grundsätzlich reziprok vorzugehen. Zusätzlich wird bei den spezifizierten Signaturmethoden jedoch – sofern sinnvoll und notwendig – Hinweise und Anwendungsnotizen für die Signaturverifikation angegeben.

4.1 Textuelle Signatur, Version 1.0.0

4.1.1 Charakteristik

Methoden-Kennzeichnung: `urn:pdfsigfilter:bka.gv.at:text:v1.0.0`

Input-Datenstrom: das zu signierende PDF-Dokument
(binärer Datenstrom, application/pdf)

Signierter Datenstrom: der aus dem PDF-Dokument extrahierte Text
(binärer Datenstrom, text/plain)

Art der Signatur: XML-Signatur, Enveloping Signature

Zulässige Signaturparameter: Default (MOA), Default (BKU), `etsi-bka-1.0`

Anwendbarkeit: NICHT EMPFOHLEN
deprecated, wurde ersetzt durch
`urn:pdfsigfilter:bka.gv.at:text:v1.1.0`

4.1.2 Aufbereitung der zu signierenden Daten

Der Aufbereitungsprozess ist aus Sicht des Signaturerstellungsprozesses definiert. Bei der Verifikation ist analog vorzugehen (siehe auch Hinweis in Abschnitt 4.1.5).

Der Input-Datenstrom MUSS wie folgt behandelt werden:

1. Der Input-Datenstrom (das PDF-Dokument) wird geöffnet.
2. Es wird der Text des gegebenen Originaldokuments extrahiert. Dabei MÜSSEN folgende Vorgaben beachtet werden:
    a. der extrahierte Text MUSS eine Zeichenfolge sein, die den auf dem PDF-Dokument dargestellten Text entspricht.
    b. die Zeichenfolge MUSS der Leserichtung folgend von links oben nach rechts unten aufgelöst werden.
    c. Besonderheiten der PDF-Repräsentation, wie etwa die Darstellung fett gedruckter Text-Teile durch Überlappung leicht versetzter Einzelzeichen, MÜSSEN ignoriert und auf den eigentliche Textinhalt reduziert werden.
3. Auf den extrahierten Text MÜSSEN die folgenden Normalisierungsmaßnahmen in der hier festgelegten Reihenfolge angewendet werden:
    a. Alle NULL-Zeichen (\u0000) werden entfernt.
    b. Alle Tabulatoren (\u0009) und Seitenumbrüche (\u000C) werden durch einzelne Leerzeichen (\u0020) ersetzt.
    c. Alle No-Break Spaces (\u00A0) werden durch Leerzeichen (\u0020) ersetzt.
    d. Alle Vorkommnisse von Zeilenumbrüchen (Newlines) – systemabhängig, zum Beispiel bei Windows die Kombination der Zeichen Zeilenumbruch (\u000D) und Zeilenvorschub (\u000A) bzw. bei MacOS nur das Zeichen Zeilenvorschub (\u000A) – werden durch ein Zeichen Zeilenvorschub (\u000A) ersetzt.
    e. Mehrfache Zeilenumbrüche, das heißt zwei oder mehrere, werden auf zwei Zeilenumbrüche (zwei Zeichen \u000A) reduziert.
    f. Alle mehrfachen Leerzeichen (\u0020) werden durch ein einfaches Leerzeichen (\u0020) ersetzt.
    g. Leerzeichen (\u0020) am Zeilenanfang oder am Zeilenende werden entfernt.
    h. Leerzeilen, das sind Zeilen ohne jeglichen Inhalt bzw. die nur mehr ein Leerzeichen enthalten, am Anfang bzw. am Ende des gesamten Textes (Dokuments) werden entfernt.
    i. Alle Arten von Apostrophen (Zeichen wie \u0060, \u00B4, \u2018, \u2019, \u201A, \u201B) werden durch das Zeichen Apostroph (\u0027) ersetzt.
    j. Alle Arten von Anführungsstriche (Zeichen wie \u201C, \u201D, \u201E, \u201F) werden durch das Zeichen Anführungszeichen (\u0022) ersetzt.
    k. Alle Arten von Bindestriche (Zeichen wie \u00AD, \u2013, \u2014) werden durch das Zeichen Bindestrich (\u002D) ersetzt.

Der resultierende Datenstrom repräsentiert den aus dem PDF-Dokument (Input-Datenstrom) extrahierten Text in Form von Unicode-Zeichen. Dieser Datenstrom wird signiert.

Der MIME-Type des zu signierenden Datenstroms MUSS im Rahmen der XML-Signatur auf `text/plain` gesetzt werden. Dementsprechend MUSS in den erstellten XML-Signaturen, sofern diese Angaben zu Eigenschaften des signierten Dokumentes beinhalten (z.B. durch das Element `etsi:SignedDataObjectProperties/etsi:DataObjectFormat`) enthalten, der MIME-Type mit `text/plain` angegeben werden.

4.1.3 XML-Signaturformat

Die resultierende Signatur ist eine XML Signatur nach [4]. Die zu signierenden Daten MÜSSEN nach Aufbereitung ohne weitere Veränderung als zu signierenden Daten für die Bildung der XML-Signatur herangezogen werden.

Der Transformationspfad MUSS die folgenden Transformationen in dieser Reihenfolge enthalten:

1. Base-64 Transformation der zu signierenden Daten (Algorithmus-Identifier `http://www.w3.org/2000/09/xmldsig#base64`)

Die zu erstellende XML-Signatur ist eine Enveloping Signature gem. [4], welche in Form eines Datenobjekts die signierten Daten eingebettet enthält. Diese MÜSSEN Base-64 kodiert als `dsig:Object` Element in die XML-Signatur eingebettet werden (näheres dazu siehe [4] und [6]).

Die erstellte XML-Signatur folgt den Vorgaben des österreichischen E-Governments bzw. den Vorgaben für XML-Signaturen aus der Spezifikation der österreichischen Bürgerkarte (siehe [6]).

Beispiel einer XML-Signatur nach diesen Vorgaben (erstellt mit der Bürgerkartensoftware IT-Solution trustDesk basic):

```
<dsig:Signature Id="signature-1161003152-26578093-24674"
xmlns:dsig="http://www.w3.org/2000/09/xmldsig#">
  <dsig:SignedInfo>
    <dsig:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-
20010315"/>
    <dsig:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#ecdsa-
sha1"/>
    <dsig:Reference Id="signed-data-reference-0-1161003152-26578093-5873" URI="#signed-
data-object-0-1161003152-26578093-8480">
      <dsig:Transforms>
        <dsig:Transform Algorithm="http://www.w3.org/2002/06/xmldsig-filter2">
          <xpf:XPath Filter="intersect" xmlns:xpf="http://www.w3.org/2002/06/xmldsig-
filter2">id(&apos;signed-data-object-0-1161003152-26578093-
8480&apos;)/node()</xpf:XPath>
        </dsig:Transform>
        <dsig:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#base64"/>
      </dsig:Transforms>
      <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
      <dsig:DigestValue>HASH-WERT DER 1. REFERENZ</dsig:DigestValue>
    </dsig:Reference>
    <dsig:Reference Id="etsi-data-reference-0-1161003152-26578093-26221"
Type="http://uri.etsi.org/01903/v1.1.1#SignedProperties" URI="#etsi-data-object-0-
1161003152-26578093-25255">
      <dsig:Transforms>
        <dsig:Transform Algorithm="http://www.w3.org/2002/06/xmldsig-filter2">
          <xpf:XPath Filter="intersect" xmlns:xpf="http://www.w3.org/2002/06/xmldsig-
filter2">id(&apos;etsi-data-object-0-1161003152-26578093-
25255&apos;)/node()</xpf:XPath>
        </dsig:Transform>
      </dsig:Transforms>
      <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
      <dsig:DigestValue>HASH-WERT DER 2. REFERENZ</dsig:DigestValue>
    </dsig:Reference>
  </dsig:SignedInfo>
  <dsig:SignatureValue>SIGNATURWERT</dsig:SignatureValue>
  <dsig:KeyInfo>
    <dsig:X509Data>
      <dsig:X509Certificate>ZERTIFIKAT</dsig:X509Certificate>
    </dsig:X509Data>
  </dsig:KeyInfo>
  <dsig:Object Id="signed-data-object-0-1161003152-26578093-8480">
    <sl:Base64Content>SIGNIERTE DATEN (BASE64)</sl:Base64Content>
  </dsig:Object>
```

```
 <dsig:Object Id="etsi-data-object-0-1161003152-26578093-25255">
  <etsi:QualifyingProperties Target="#signature-1161003152-26578093-24674"
xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
xmlns:etsi="http://uri.etsi.org/01903/v1.1.1#">
   <etsi:SignedProperties>
    <etsi:SignedSignatureProperties>
     <etsi:SigningTime>SIGNATURZEITPUNKT</etsi:SigningTime>
     <etsi:SigningCertificate>
      <etsi:Cert>
       <etsi:CertDigest>
        <etsi:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
        <etsi:DigestValue>HASHWERT DES SIGNATURZERTIFIKATES</etsi:DigestValue>
       </etsi:CertDigest>
       <etsi:IssuerSerial>
        <dsig:X509IssuerName>AUSSTELLER DES ZERTIFIKATS</dsig:X509IssuerName>
        <dsig:X509SerialNumber>SERIENNUMMER DES ZERTIFIKATS</dsig:X509SerialNumber>
       </etsi:IssuerSerial>
      </etsi:Cert>
     </etsi:SigningCertificate>
     <etsi:SignaturePolicyIdentifier>
      <etsi:SignaturePolicyImplied/>
     </etsi:SignaturePolicyIdentifier>
    </etsi:SignedSignatureProperties>
    <etsi:SignedDataObjectProperties>
     <etsi:DataObjectFormat ObjectReference="#signed-data-reference-0-1161003152-
26578093-5873">
      <etsi:MimeType>text/plain</etsi:MimeType>
     </etsi:DataObjectFormat>
    </etsi:SignedDataObjectProperties>
   </etsi:SignedProperties>
  </etsi:QualifyingProperties>
 </dsig:Object>
</dsig:Signature>
```

Dieses Beispiel enthält einige Besonderheiten der Signaturerstellungskomponente (Bürgerkartensoftware und Signaturerstellungseinheit), auf die in Verbindung mit Signaturparametern noch eingegangen wird. Aus Gründen der Übersichtlichkeit wurden variable Inhalte größtenteils durch verbale Umschreibungen ersetzt (eingerahmter Text).

4.1.4 Einbettung der Signatur in das PDF-Dokument

Die resultierende XML-Signatur MUSS in Form der in Abschnitt 3 definierten Repräsentation in das PDF-Dokument integriert werden. Die eingebrachte Signatur-Repräsentation DARF KEINE Zeichen und Elemente enthalten, die im Zuge der Verifikation nicht wieder entfernt werden können und so die Verifikation verhindern.

Die Einbettung der Signatur-Repräsentation im PDF-Dokument KANN mit Hilfe eines Inkrementellen Update Blocks (Incremental Update Block, Abschnitt 3.4.5 in [2]) realisiert werden. Der Text der eingebetteten Signatur-Repräsentation MUSS im vom signierten PDF-Dokument extrahierten Text enthalten sein.

Die Signatur-Repräsentation MUSS auch im extrahierten Text entsprechend der Leserichtung an der korrespondierenden Stelle vorkommen.

4.1.5 Anwendungshinweis zur Verifikation

Zur Verifikation von derart signierten Dokumenten MUSS reziprok zu der in dieser Spezifikation festgelegten Vorgehensweise verfahren werden. Zusätzlich werden die folgenden Anwendungshinweise gegeben.

Gegeben sei ein unter Anwendung der hier spezifizierten Signaturmethode textuell signiertes PDF-Dokument. Die Applikation MUSS aus der in der Signatur-Repräsentation enthaltenen Methoden-Kennung das korrekte Signaturverfahren bestimmen und somit das adäquate Verifikationsverfahren anwenden.

Die Vorgehensweise der Verifikation im Überblick:

1. Der gesamte Dokumenttext des zu prüfenden PDF-Dokuments wird extrahiert (analog dem Vorgehen bei Signaturerstellung (vgl. die Vorschrift zur Aufbereitung der zu signierenden Daten).
2. Im extrahierten Dokumenttext befindet sich die textuelle Repräsentation des Signaturblocks. Dieser wird herausgelöst und aus dem Text entfernt. Dadurch wird der ursprünglich signierte Text gewonnen. Dies entspricht dem signierten Datenstrom.
3. Entsprechend den Vorgaben dieser Art der textuellen Signatur werden die Signaturattribute, wie Signaturwert, Datum etc., sowie gegebenenfalls angegebene Signaturparameter (sowie die Kennzeichnung des Signaturparameter-Profils) aus der herausgelösten Textrepräsentation des Signaturblocks extrahiert. Die so gewonnenen Daten werden zur technischen Rekonstruktion der XML-Signatur benötigt.
4. Die dem signierten PDF-Dokument hinterlegte XML-Signatur wird anhand der zuvor gewonnenen Daten unter Berücksichtigung des jeweiligen Signaturparameter-Profils, bzw. unter Anwendung des damit festgelegten XML-Signaturlayouts, rekonstruiert.
5. Die rekonstruierte XML-Signatur wird verifiziert.

4.2 Textuelle Signatur, Version 1.1.0

4.2.1 Charakteristik

Methoden-Kennzeichnung:	`urn:pdfsigfilter:bka.gv.at:text:v1.1.0`
Input-Datenstrom:	das zu signierende PDF-Dokument (binärer Datenstrom, application/pdf)
Signierte Datenstrom:	der aus dem PDF-Dokument extrahierte Text (binärer Datenstrom, text/plain)
Art der Signatur:	XML-Signatur, Detached Signature
Zulässige Signaturparameter:	keine Einschränkung
Anwendbarkeit:	EMPFOHLEN

4.2.2 Aufbereitung der zu signierenden Daten

Der Aufbereitungsprozess ist aus Sicht des Signaturerstellungsprozesses definiert. Bei der Verifikation ist analog vorzugehen (siehe auch Hinweis in Abschnitt 4.2.5).

Der Input-Datenstrom MUSS wie folgt behandelt werden:

1. Der Input-Datenstrom (das PDF-Dokument) wird geöffnet.
2. Es wird der Text des gegebenen Originaldokuments extrahiert. Dabei MÜSSEN folgende Vorgaben beachtet werden:
   a. der extrahierte Text MUSS eine Zeichenfolge sein, die den auf dem PDF-Dokument dargestellten Text entspricht.
   b. die Zeichenfolge MUSS der Leserichtung folgend von links oben nach rechts unten aufgelöst werden.
   c. Besonderheiten der PDF-Repräsentation, wie etwa die Darstellung fett gedruckter Text-Teile durch Überlappung leicht versetzter Einzelzeichen, MÜSSEN ignoriert und auf den eigentliche Textinhalt reduziert werden.

3. Auf den extrahierten Text MÜSSEN die folgenden Normalisierungsmaßnahmen in der hier festgelegten Reihenfolge angewendet werden:
   a. Alle NULL-Zeichen (\u0000) werden entfernt.
   b. Alle Tabulatoren (\u0009) und Seitenumbrüche (\u000C) werden durch einzelne Leerzeichen (\u0020) ersetzt.
   c. Alle No-Break Spaces (\u00A0) werden durch Leerzeichen (\u0020) ersetzt.
   d. Alle Vorkommnisse von Zeilenumbrüchen (Newlines) – systemabhängig, zum Beispiel bei Windows die Kombination der Zeichen Zeilenumbruch (\u000D) und Zeilenvorschub (\u000A) bzw. bei MacOS nur das Zeichen Zeilenvorschub (\u000A) – werden durch ein Zeichen Zeilenvorschub (\u000A) ersetzt.
   e. Mehrfache Zeilenumbrüche, das heißt zwei oder mehrere, werden auf zwei Zeilenumbrüche (zwei Zeichen \u000A) reduziert.
   f. Alle mehrfachen Leerzeichen (\u0020) werden durch ein einfaches Leerzeichen (\u0020) ersetzt.
   g. Leerzeichen (\u0020) am Zeilenanfang oder am Zeilenende werden entfernt.
   h. Leerzeilen, das sind Zeilen ohne jeglichen Inhalt bzw. die nur mehr ein Leerzeichen enthalten, am Anfang bzw. am Ende des gesamten Textes (Dokuments) werden entfernt.
   i. Alle Arten von Apostrophen (Zeichen wie \u0060, \u00B4, \u2018, \u2019, \u201A, \u201B) werden durch das Zeichen Apostroph (\u0027) ersetzt.
   j. Alle Arten von Anführungsstriche (Zeichen wie \u201C, \u201D, \u201E, \u201F) werden durch das Zeichen Anführungszeichen (\u0022) ersetzt.
   k. Alle Arten von Bindestriche (Zeichen wie \u00AD, \u2013, \u2014) werden durch das Zeichen Bindestrich (\u002D) ersetzt.

Der resultierende Datenstrom repräsentiert den aus dem PDF-Dokument (Input-Datenstrom) extrahierten Text in Form von Unicode-Zeichen. Dieser Datenstrom wird signiert.

Der MIME-Type des zu signierenden Datenstroms MUSS im Rahmen der XML-Signatur auf `text/plain` gesetzt werden. Dementsprechend MUSS in den erstellten XML-Signaturen, sofern diese Angaben zu Eigenschaften des signierten Dokumentes beinhalten (z.B. durch das Element `etsi:SignedDataObjectProperties/etsi:DataObjectFormat`) enthalten, der MIME-Type mit `text/plain` angegeben werden.

4.2.3 XML-Signaturformat

Die resultierende Signatur ist eine XML Signatur nach [4]. Die zu signierenden Daten MÜSSEN nach Aufbereitung ohne weitere Veränderung als zu signierenden Daten für die Bildung der XML-Signatur herangezogen werden.

Die zu erstellende XML-Signatur MUSS eine Detached Signature gem. [4] sein.

Die erstellte XML-Signatur folgt den Vorgaben des österreichischen E-Governments bzw. den Vorgaben für XML-Signaturen aus der Spezifikation der österreichischen Bürgerkarte (siehe [6]).

Beispiel einer XML-Signatur nach diesen Vorgaben (erstellt mit der Bürgerkartensoftware IT-Solution trustDesk basic):

```
<dsig:Signature xmlns:dsig="http://www.w3.org/2000/09/xmldsig#" Id="signature-
1201005938-3018328-13744">
  <dsig:SignedInfo>
    <dsig:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-
20010315"/>
    <dsig:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#ecdsa-
sha1"/>
    <dsig:Reference Id="signed-data-reference-0-1201005938-3018328-20478"
URI="urn:Document">
      <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
      <dsig:DigestValue>HASH-WERT DER 1. REFERENZ</dsig:DigestValue>
    </dsig:Reference>
    <dsig:Reference Id="etsi-data-reference-0-1201005938-3018328-327"
Type="http://uri.etsi.org/01903/v1.1.1#SignedProperties"
URI="#xmlns(etsi=http://uri.etsi.org/01903/v1.1.1%23)%20xpointer(id('etsi-data-object-
0-1201005938-3018328-
18413')/child::etsi:QualifyingProperties/child::etsi:SignedProperties)">
      <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
      <dsig:DigestValue>HASH-WERT DER 2. REFERENZ</dsig:DigestValue>
    </dsig:Reference>
  </dsig:SignedInfo>
  <dsig:SignatureValue>SIGNATURWERT</dsig:SignatureValue>
  <dsig:KeyInfo>
    <dsig:X509Data>
      <dsig:X509Certificate>ZERTIFIKAT</dsig:X509Certificate>
    </dsig:X509Data>
  </dsig:KeyInfo>
  <dsig:Object Id="etsi-data-object-0-1201005938-3018328-18413">
    <etsi:QualifyingProperties xmlns:etsi="http://uri.etsi.org/01903/v1.1.1#"
Target="#signature-1201005938-3018328-13744">
      <etsi:SignedProperties>
        <etsi:SignedSignatureProperties>
          <etsi:SigningTime>SIGNATURZEITPUNKT</etsi:SigningTime>
          <etsi:SigningCertificate>
            <etsi:Cert>
              <etsi:CertDigest>
                <etsi:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
                <etsi:DigestValue>HASH-WERT DES ZERTIFIKATS</etsi:DigestValue>
              </etsi:CertDigest>
              <etsi:IssuerSerial>
                <dsig:X509IssuerName>AUSSTELLER DES ZERTIFIKATS</dsig:X509IssuerName>
                <dsig:X509SerialNumber>SERIENNUMMER DES ZERTIFIKATS</dsig:X509SerialNumber>
              </etsi:IssuerSerial>
            </etsi:Cert>
          </etsi:SigningCertificate>
          <etsi:SignaturePolicyIdentifier>
            <etsi:SignaturePolicyImplied/>
          </etsi:SignaturePolicyIdentifier>
        </etsi:SignedSignatureProperties>
        <etsi:SignedDataObjectProperties>
          <etsi:DataObjectFormat ObjectReference="#signed-data-reference-0-1201005938-
3018328-20478">
            <etsi:MimeType>text/plain</etsi:MimeType>
          </etsi:DataObjectFormat>
        </etsi:SignedDataObjectProperties>
      </etsi:SignedProperties>
    </etsi:QualifyingProperties>
  </dsig:Object>
</dsig:Signature>
```

Dieses Beispiel enthält einige Besonderheiten der Signaturerstellungskomponente (Bürgerkartensoftware und Signaturerstellungseinheit), auf die in Verbindung mit Signaturparametern noch eingegangen wird. Aus Gründen der Übersichtlichkeit wurden variable Inhalte größtenteils durch verbale Umschreibungen ersetzt (eingerahmter Text).

4.2.4 Einbettung der Signatur in das PDF-Dokument

Die resultierende XML-Signatur MUSS in Form der in Abschnitt 3 definierten Repräsentation in das PDF-Dokument integriert werden. Die eingebrachte Signatur-Repräsentation DARF KEINE Zeichen und Elemente enthalten, die im Zuge der Verifikation nicht wieder entfernt werden können und so die Verifikation verhindern.

Die Einbettung der Signatur-Repräsentation im PDF-Dokument MUSS mit Hilfe eines Inkrementellen Update Blocks (Incremental Update Block, Abschnitt 3.4.5 in [2]) realisiert werden. Der Text der eingebetteten Signatur-Repräsentation MUSS im vom signierten PDF-Dokument extrahierten Text enthalten sein.

Die Signatur-Repräsentation MUSS auch im extrahierten Text entsprechend der Leserichtung an der korrespondierenden Stelle vorkommen.

4.2.5 Anwendungshinweis zur Verifikation

Zur Verifikation von derart signierten Dokumenten MUSS reziprok zu der in dieser Spezifikation festgelegten Vorgehensweise verfahren werden. Zusätzlich werden die folgenden Anwendungshinweise gegeben.

Gegeben sei ein unter Anwendung der hier spezifizierten Signaturmethode textuell signiertes PDF-Dokument. Die Applikation MUSS aus der in der Signatur-Repräsentation enthaltenen Methoden-Kennung das korrekte Signaturverfahren bestimmen und somit das adäquate Verifikationsverfahren anwenden.

Die Vorgehensweise der Verifikation im Überblick:

1. Der gesamte Dokumenttext des zu prüfenden PDF-Dokuments wird extrahiert (analog dem Vorgehen bei Signaturerstellung (vgl. die Vorschrift zur Aufbereitung der zu signierenden Daten).
2. Im extrahierten Dokumenttext befindet sich die textuelle Repräsentation des Signaturblocks. Dieser wird herausgelöst und aus dem Text entfernt. Dadurch wird der ursprünglich signierte Text gewonnen. Dies entspricht dem signierten Datenstrom.
3. Entsprechend den Vorgaben dieser Art der textuellen Signatur werden die Signaturattribute, wie Signaturwert, Datum etc., sowie gegebenenfalls angegebene Signaturparameter (sowie die Kennzeichnung des Signaturparameter-Profils) aus der herausgelösten Textrepräsentation des Signaturblocks extrahiert. Die so gewonnenen Daten werden zur technischen Rekonstruktion der XML-Signatur benötigt.
4. Die dem signierten PDF-Dokument hinterlegte XML-Signatur wird anhand der zuvor gewonnenen Daten unter Berücksichtigung des jeweiligen Signaturparameter-Profils, bzw. unter Anwendung des damit festgelegten XML-Signaturlayouts, rekonstruiert.
5. Die rekonstruierte XML-Signatur wird verifiziert.

4.3 Binäre Signatur, Version 1.0.0

4.3.1 Charakteristik

Methoden-Kennzeichnung:	`urn:pdfsigfilter:bka.gv.at:binaer:v1.0.0`
Input-Datenstrom:	das zu signierende PDF-Dokument (binärer Datenstrom, application/pdf)
Signierter Datenstrom:	das aufbereitete PDF-Dokument (binärer Datenstrom, application/pdf)
Art der Signatur:	XML-Signatur, Enveloping Signature
Zulässige Signaturparameter:	Default (MOA), Default (BKU), `etsi-bka-1.0`
Anwendbarkeit:	NICHT EMPFOHLEN deprecated, wurde ersetzt durch `urn:pdfsigfilter:bka.gv.at:binaer:v1.1.0`

4.3.2 Aufbereitung der zu signierenden Daten

Der Aufbereitungsprozess ist aus Sicht des Signaturerstellungsprozesses definiert. Bei der Verifikation ist analog vorzugehen (siehe auch Hinweis in Abschnitt 4.3.5).

Die Binäre Signatur sieht vor, dass das gesamte PDF-Dokument binär signiert wird.

Um Manipulationen an einer Binären Signatur auszuschließen, MUSS das Dokument selbst mit samt der vorbereiteten Signatur-Repräsentation (gemäß Vorgaben aus Abschnitt 3) signiert werden. Lediglich die im Zuge der Signaturerstellung gewonnenen Informationen – Signaturwert, Signaturzeitpunkt, Angaben zum Signaturzertifikat bzw. die Signaturattribute der erstellten XML-Signatur im Allgemeinen – MÜSSEN nach der Signaturprozedur in das signierte und vorbereitete PDF-Dokument eingefügt werden. Im Zuge der Signaturprüfung MÜSSEN die nach der Signaturerstellung eingebetteten Werte wieder durch die zum Signaturzeitpunkt verwendeten „Platzhaltern“ ersetzt werden. Dies entspricht somit wieder dem signierten Dokument.

Das binäre signierte PDF-Dokument MUSS zur Signatur vorbereitet werden; dazu MÜSSEN die folgenden Schritte angewendet werden:

1. Dem PDF-Dokument MUSS die Signatur-Repräsentation (Signaturblock) bereits vor der Signaturerstellung eingebettet werden. Dazu ist gemäß den Vorgaben aus Abschnitt 4.3.4 der Signaturblock erstellt und in das Dokument eingebrachte werden. Anstelle der zu diesem Zeitpunkt noch unbekannten Werte (Signaturattribute wie Signaturwert, Signaturzeitpunkt, Angaben zum Signaturzertifikat, etc.) MÜSSEN durch semantisch wertfreie Füllzeichen ersetzt werden. Als Füllzeichen MUSS das NULL-Byte (numerisch 0) verwendet werden (dies ist das Default Füllzeichen für die in der Signatur-Repräsentation vorgesehenen Wertebereiche zur Fassung der Signaturattribute; siehe dazu auch die Vorgaben aus Abschnitt 4.3.4.1.6). Nach erfolgter Signatur werden gemäß den Vorgaben aus Abschnitt 4.3.4 die Signaturattribute in den durch Füllzeichen vorbereiteten Wertebereiche der Signatur-Repräsentation eingefüllt.

Das so vorbereitete PDF-Dokument wird als binärer Datenstrom (Octet Stream) interpretiert und als Datenstrom für die Signaturerstellung herangezogen. Dieser Datenstrom wird signiert.

Diese Signaturmethode wurde auch zur Verwendung mit einer frühen Version der Bürgerkartensoftware definiert. Daher MUSS der zu signierende, binäre Datenstrom explizit Base64-kodiert und im Zuge der Signaturerstellung als Text interpretiert werden. Der MIME-Type des zu signierenden Datenstroms MUSS daher im Rahmen der XML-Signatur auf `text/plain` gesetzt werden. Dementsprechend MUSS in den erstellten XML-Signaturen, sofern diese Angaben zu Eigenschaften des signierten Dokumentes beinhalten (z.B. durch das Element `etsi:SignedDataObjectProperties/etsi:DataObjectFormat`) enthalten, der MIME-Type mit `text/plain` angegeben werden.

4.3.3 XML-Signaturformat

Die resultierende Signatur ist eine XML Signatur nach [4]. Die zu signierenden Daten MÜSSEN nach Aufbereitung ohne weitere Veränderung als zu signierenden Daten für die Bildung der XML-Signatur herangezogen werden.

Der Transformationspfad MUSS die folgenden Transformationen in dieser Reihenfolge enthalten:

1. Base64 Transformation der zu signierenden Daten (Algorithmus-Identifier `http://www.w3.org/2000/09/xmldsig#base64`)

Die zu erstellende XML-Signatur ist eine Enveloping Signature gem. [4], welche in Form eines Datenobjekts die signierten Daten eingebettet enthält. Diese MÜSSEN Base64-kodiert als dsig:Object Element in die XML-Signatur eingebettet werden (näheres dazu siehe [4] und [6]). Durch diese explizite Base64-Transformation kann der zu signierende binäre Datenstrom als Text interpretiert werden.

Die erstellte XML-Signatur folgt den Vorgaben des österreichischen E-Governments bzw. den Vorgaben für XML-Signaturen aus der Spezifikation der österreichischen Bürgerkarte (siehe [6]).

Beispiel einer XML-Signatur nach diesen Vorgaben (erstellt mit der Bürgerkartensoftware IT-Solution trustDesk basic):

```
<dsig:Signature Id="signature-1161003152-26578093-24674"
xmlns:dsig="http://www.w3.org/2000/09/xmldsig#">
  <dsig:SignedInfo>
    <dsig:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-
20010315"/>
    <dsig:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#ecdsa-
sha1"/>
    <dsig:Reference Id="signed-data-reference-0-1161003152-26578093-5873" URI="#signed-
data-object-0-1161003152-26578093-8480">
      <dsig:Transforms>
        <dsig:Transform Algorithm="http://www.w3.org/2002/06/xmldsig-filter2">
          <xpf:XPath Filter="intersect" xmlns:xpf="http://www.w3.org/2002/06/xmldsig-
filter2">id(&apos;signed-data-object-0-1161003152-26578093-
8480&apos;)/node()</xpf:XPath>
        </dsig:Transform>
        <dsig:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#base64"/>
      </dsig:Transforms>
      <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
      <dsig:DigestValue>HASH-WERT DER 1. REFERENZ</dsig:DigestValue>
    </dsig:Reference>
    <dsig:Reference Id="etsi-data-reference-0-1161003152-26578093-26221"
Type="http://uri.etsi.org/01903/v1.1.1#SignedProperties" URI="#etsi-data-object-0-
1161003152-26578093-25255">
      <dsig:Transforms>
        <dsig:Transform Algorithm="http://www.w3.org/2002/06/xmldsig-filter2">
          <xpf:XPath Filter="intersect" xmlns:xpf="http://www.w3.org/2002/06/xmldsig-
filter2">id(&apos;etsi-data-object-0-1161003152-26578093-
25255&apos;)/node()</xpf:XPath>
        </dsig:Transform>
      </dsig:Transforms>
```

```
    <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
    <dsig:DigestValue>HASH-WERT DER 2. REFERENZ</dsig:DigestValue>
   </dsig:Reference>
  </dsig:SignedInfo>
  <dsig:SignatureValue>SIGNATURWERT</dsig:SignatureValue>
  <dsig:KeyInfo>
   <dsig:X509Data>
    <dsig:X509Certificate>ZERTIFIKAT</dsig:X509Certificate>
   </dsig:X509Data>
  </dsig:KeyInfo>
  <dsig:Object Id="signed-data-object-0-1161003152-26578093-8480">
   <sl:Base64Content>SIGNIERTE DATEN (BASE64)</sl:Base64Content>
  </dsig:Object>
  <dsig:Object Id="etsi-data-object-0-1161003152-26578093-25255">
   <etsi:QualifyingProperties Target="#signature-1161003152-26578093-24674"
xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
xmlns:etsi="http://uri.etsi.org/01903/v1.1.1#">
    <etsi:SignedProperties>
     <etsi:SignedSignatureProperties>
      <etsi:SigningTime>SIGNATURZEITPUNKT</etsi:SigningTime>
      <etsi:SigningCertificate>
       <etsi:Cert>
        <etsi:CertDigest>
         <etsi:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
         <etsi:DigestValue>HASH-WERT DES SIGNATURZERTIFIKATS</etsi:DigestValue>
        </etsi:CertDigest>
        <etsi:IssuerSerial>
         <dsig:X509IssuerName>AUSSTELLER DES ZERTIFIKATS</dsig:X509IssuerName>
         <dsig:X509SerialNumber>SERIENNUMMER DES ZERTIFIKATS</dsig:X509SerialNumber>
        </etsi:IssuerSerial>
       </etsi:Cert>
      </etsi:SigningCertificate>
      <etsi:SignaturePolicyIdentifier>
       <etsi:SignaturePolicyImplied/>
      </etsi:SignaturePolicyIdentifier>
     </etsi:SignedSignatureProperties>
     <etsi:SignedDataObjectProperties>
      <etsi:DataObjectFormat ObjectReference="#signed-data-reference-0-1161003152-
26578093-5873">
       <etsi:MimeType>text/plain</etsi:MimeType>
      </etsi:DataObjectFormat>
     </etsi:SignedDataObjectProperties>
    </etsi:SignedProperties>
   </etsi:QualifyingProperties>
  </dsig:Object>
</dsig:Signature>
```

Dieses Beispiel enthält einige Besonderheiten der Signaturerstellungskomponente (Bürgerkartensoftware und Signaturerstellungseinheit), auf die in Verbindung mit Signaturparametern noch eingegangen wird. Aus Gründen der Übersichtlichkeit wurden variable Inhalte größtenteils durch verbale Umschreibungen ersetzt (eingerahmter Text).

4.3.4 Einbettung der Signatur in das PDF-Dokument

Die resultierende XML-Signatur MUSS letztlich in Form der in Abschnitt 3 definierten Repräsentation in das PDF-Dokument integriert werden.

Die Einbettung der Signatur-Repräsentation (Signaturblock) im PDF-Dokument MUSS mit Hilfe eines Inkrementellen Update Blocks (Incremental Update Block, Abschnitt 3.4.5 in [2]) realisiert werden. Dieser Block MUSS folgende Struktur aufweisen:

1. Dieser Incremental Update Block MUSS ein eigenes Dictionary, das EGIZ-Dictionary (siehe 4.3.4.1), enthalten. Dieses MUSS ein indirektes Objekt sein.
2. Der gesamte sichtbare Signaturblock MUSS in ein XObject Form eingebettet sein. (siehe /SigXObject Key des EGIZ Dictionaries, Abschnitt 4.3.4.1)
3. Das trailer-Dictionary des Incremental Update Blocks MUSS einen Key /EGIZSigDict enthalten. Wert dieses Keys MUSS eine indirekte Referenz auf das EGIZ-Dictionary sein.

Der nachfolgende Abschnitt beschreibt das EGIZ-Dictionary im Detail.

Als Spezifikum dieses Algorithmus MUSS die Signatur-Repräsentation bereits vor dem Signaturvorgang, im Zuge der Aufbereitung der zu signierenden Daten, in das PDF-Dokument eingebracht werden. Anstelle der zu diesem Zeitpunkt noch unbekannten Werte, wie bspw. Signaturattribute (das sind zum Beispiel Signaturwert, Signaturzeitpunkt, Angaben zum Signaturzertifikat, etc.), MÜSSEN die dafür vorgesehenen Wertebereiche mit semantisch wertfreien Füllzeichen aufgefüllt werden.

Das mit diesem vorbereiteten aber leeren Signaturblock versehene PDF-Dokument wird in seiner binären Repräsentation elektronisch signiert. Nach dem Signaturprozess MÜSSEN die dabei ermittelten Werte (Signaturattribute) in die dafür vorgesehenen Wertebereiche der Signatur-Repräsentation (in den nachfolgenden Abschnitten auch als "Aussparung" bezeichnet) eingefüllt werden.

4.3.4.1 EGIZ-Dictionary

Die Keys des EGIZ Dictionaries MÜSSEN, falls verwendet, in folgender Reihenfolge vorhanden sein:

Key	Kardinalität	Beschreibung
/Type	MUSS	Bezeichnet den Typ des Dictionaries. Es MUSS /EGIZSigDict lauten.
/ODS	MUSS	Enthält die Größe des gesamten Dokuments (Originaldokument inklusive Incremental Update Block für die Signatur-Repräsentation). Alle Byte-Ranges MÜSSEN innerhalb dieses Werts liegen.
/ID	MUSS	Enthält die Byte-Range der Zeichenkette (String), die die angewendete Signaturmethode (Methode) identifiziert. Siehe Abschnitt 4.3.4.1.1.
/SigXObject	MUSS	Enthält die indirekte Referenz auf das XObject Form der Signatur-Repräsentation. Diese MUSS eine indirekte Referenz sein.

Key	Kardinalität	Beschreibung
/ByteRange	MUSS	Enthält aufsteigend sortierte Byte-Ranges, wodurch die signierten Bereiche des Dokumentes auf binärer Ebene identifiziert werden. Dieser Array ist analog zu den Byte-Range Arrays der Adobe PDF Signaturen definiert (siehe PDF Reference 1.6 [3], Kapitel 8.7, Tabelle 8.98). Die erste Byte-Range MUSS an Position 0 beginnen und die letzte Byte-Range dieses Arrays MUSS das Ende des Dokumentes referenzieren. Siehe Abschnitt 4.3.4.1.2.
/replaces	MUSS	Dieses Feld gibt an, welche Elemente der Signatur-Repräsentation nach der Signaturerstellung durch signaturspezifische Werte ersetzt werden müssen. Das Feld stellt ein PDF Array von PDF-Names dar, welche angeben, welches Signaturattribut in den Aussparungen der binären Signatur – binär repräsentiert durch die inversen Byte-Ranges des /ByteRange-Elements – enthalten sind. Gibt es in einem Dokument n Byte-Ranges, so enthält das /replaces-Array (n-1) Einträge für die (n-1) Bereiche. Siehe Abschnitt 4.3.4.1.3.
/encodings	MUSS	Das Feld stellt ein PDF Array von PDF-Names dar, welche angeben, welches Encoding bei den Daten der Signaturattribute angewendet wurde. Die Reihenfolge der in diesem Array angegebenen Encodings MUSS mit der Reihenfolge des /replaces-Arrays korrespondieren. Siehe Abschnitt 4.3.4.1.4.
/Cert	SOLL	Dieses Array von PDF-Strings KANN zur Einbettung von Zertifikatsdaten (X509-Zertifiakten) verwendet werden. Es SOLL zumindest das Signaturzertifikat selbst, es KANN jedoch auch die gesamte Zertifikatskette im Rahmen dieses Arrays als PDF-Strings eingebettet werden. Siehe Abschnitt 4.3.4.1.5.

729 Sämtliche Werte, falls nicht anders spezifiziert, MÜSSEN direkte Objekte, also direkt im EGIZ-
730 Dictionary eingebettet sein.

731 Das EGIZ-Dictionary DARF noch weitere Elemente enthalten. Diese können dazu verwendet
732 werden die Signatur mit zusätzlicher Information auszustatten.

4.3.4.1.1 /ID

Das `/ID`-Element beschreibt die Byte-Ranges (siehe Beschreibung `/ByteRange`) der Zeichenkette (String), die die angewendete Signaturmethode (Methode) identifiziert.

Der so identifizierte String ist jener Teil des Content Streams der Signatur-Repräsentation, welcher den Text der Signaturmethode enthält. Dieser String unterliegt genauso den PDF Formatierungsregeln und wird daher im Allgemeinen in mehrere PDF-Strings gebrochen sein.

Die `/ID` Byte-Ranges beschreiben den Inhalt dieser PDF-Strings des Content Streams.

Zusammengefügt MÜSSEN die durch diese Byte-Ranges spezifizierten Strings wieder den Signaturmethoden-String ergeben. Das Character Encoding des Kennzeichnungsstrings MUSS 8 bit WinAnsiEncoding sein. Zu beachten sind auch PDF String Escape Sequences (siehe Kapitel „Strings in den Löchern“).

4.3.4.1.2 /ByteRange

Statische Bereiche werden durch Byte-Ranges beschrieben. Byte-Ranges sind in Analogie zu Byte-Ranges in Adobe PDF-Signaturen definiert (siehe PDF Reference 1.6 [3], Kapitel 8.7):

- Eine Byte-Range MUSS aus dem Zahlentupel Startoffset und Länge (in Bytes) bestehen.
- Beide MÜSSEN positive Ganzzahlen sein, wobei der Startoffset auch 0 sein DARF.
- Der Startoffset MUSS vom Anfang der PDF-Datei an gemessen werden.
- Die Länge MUSS die Anzahl an Bytes ab dem Startoffset angeben, welche zur Byte-Range gehören.

In einer binären Signatur MÜSSEN alle statischen Bereiche mittels Byte-Ranges identifiziert werden. Die variablen Bereiche zwischen den Byte-Ranges werden als Aussparungen bezeichnet, da die dadurch binär identifizierten Bereiche des PDF-Dokumentes nicht von der Signatur abgedeckt werden. In die Bereiche dieser Aussparungen MÜSSEN zum Signierzeitpunkt sogenannte Platzhalter-Zeichen eingefügt sein. Nach erfolgter Signatur MÜSSEN diese durch die resultierenden Werte der Signatur (zum Beispiel Signaturwert, Signaturzeitpunkt, etc.) ersetzt werden.

Die Angaben von Byte-Ranges in den Elementen der binären Signatur MÜSSEN immer gemäß ihrer Startoffsets aufsteigend sortiert sein.

Beispiel:

Die Byte-Range (10, 5) beschreibt die Bytes an den Positionen 10, 11, 12, 13 und 14. Ein korrespondierendes `/ByteRange`-Array könnte folgendermaßen aussehen:

```
[ 0 100 110 90 ]
```

Dieses beschreibt zwei Byte-Ranges – von 0 bis 99 sowie von 110 bis 199 – und eine Aussparung – von 100 bis 109.

Mit Hilfe dieser Byte-Range Angaben werden die signierten Bereiche des Dokuments auf Byte-Ebene identifiziert. Umgekehrt werden damit jene, nichtsignierten Aussparungen identifiziert, in die nach der Signatur die Signaturattribute eingebettet werden müssen. Das nachfolgende Beispiel soll dies illustrieren.

Beispiel:

```
[…]
1 0 0 1 137.91 207 Tm
/F1 12 Tf
(123456789a)Tj
ET
BT
1 0 0 1 217.11 225 Tm
/F2 12 Tf
(123bcdefgh)Tj
1 0 0 1 217.11 213 Tm
[…]
```

Die nicht umrahmten Bereiche sind jene Bereiche, die über die Angabe von Byte-Ranges als die „signierten Bytes“ des Dokuments identifiziert werden würden. Die umrahmten Bereiche stellen hingegen Aussparungen dar, welche im signierten Dokument durch semantisch wertfreie Platzhalter ersetzt werden. Diese Struktur wird durch das `/ByteRange`-Element auf Byte-Ebene beschrieben.

4.3.4.1.3 /replaces

Das Feld stellt ein PDF Array von PDF-Names dar, welche angeben, welches Signaturattribut in den Aussparungen der binären Signatur – binär repräsentiert durch die inversen Byte-Ranges des `/ByteRange`-Elements – enthalten sind.

Es MUSS so viele Elemente im `/replaces` Array geben wie es Aussparungen durch die getroffene Definition der Byte-Ranges gibt. Gibt es in einem Dokument n Byte-Ranges, so enthält das `/replaces`-Array (n-1) Einträge für die (n-1) Aussparungen.

Jedes Element im `/replaces` Array MUSS ein gültiger PDF-Name sein, welcher den semantischen Wert (Typ) des Inhalts der Aussparung festlegt.

Folgende PDF-Namen sind zur Beschreibung der semantischen Werte definiert:

PDF-Name	Semantischer Wert (Typ) / Signaturattribut
`/nil`	Keine für die Signatur relevante Information oder der Typ des Wertes wird auf andere Art und Weise bestimmt, zum Beispiel durch eine explizite Referenz im EGIZ-Dictionary (siehe Definition des `/Cert`-Array).
`/dat`	Zeichenkette, die den Signaturzeitpunkt repräsentiert/beschreibt (oder Teilfragment dessen).
`/iss`	Zeichenkette, die den Aussteller des Signaturzertifikates repräsentiert/beschreibt (oder Teilfragment dessen).
`/snr`	Zeichenkette, die den Seriennummer des Signaturzertifikates repräsentiert/beschreibt (oder Teilfragment dessen).
`/val`	Zeichenkette, die den Signaturwert repräsentiert/beschreibt (oder Teilfragment dessen).
`/sid`	Zeichenkette, die die Signaturparameter repräsentiert/beschreibt (oder Teilfragment dessen).

Unbekannte Typen MÜSSEN im Zuge der Verifikation der Signatur als nicht spezifikationskonform zurückgewiesen werden.

Signaturattribute, bzw. die sie repräsentierenden Zeichenketten, DÜRFEN aus Platzgründen auf mehrere, aufeinanderfolgende Aussparungen aufgeteilt werden. Der Inhalt von derart aufeinander folgenden Aussparungen gleichen Typs stellt zusammengefasst den Gesamtwert des betreffenden Signaturattributs dar. Um diesen Gesamtwert zu bilden MÜSSEN die Fragmente konkateniert werden.

Das folgende Beispiel zeigt die Beschreibung der semantischen Inhalte von Aussparungen, wobei die erste Aussparung ein nicht näher spezifizierter Datenblock ist (Typ `/nil`), die zweite und die dritte Aussparung Fragmente der Zeichenkette des Signaturdatums enthalten (Typ `/dat`), die Aussparung vier bis sechs Fragmente der Zeichenkette des Signaturwertes enthalten (Typ `/val`), die siebte und achte Aussparung wiederum nicht näher spezifizierte Datenblöcke enthalten, die letzte Aussparung die die Signaturparameter beschreibende Zeichenkette repräsentiert (Typ `/sid`).

Beispiel:

```
[/nil /dat /dat /val /val /val /nil /nil /sid]
```

4.3.4.1.4 /encodings

Die Festlegung der Typen der in Aussparungen gefassten Werte durch das /replaces-Feld werden durch die analoge Festlegung von Kodierungsarten (Encoding) ergänzt. Das /encodings-Feld legt daher fest, auf welche Art die in den jeweiligen Aussparungen gefassten Daten (vor allem Zeichenketten) kodiert worden sind.

Es MUSS für jeden Wert, der in einer Aussparung nach der Signatur eingebettet und im /replaces-Feld entsprechend festgelegt wurde, spezifiziert werden, auf welche Art dieser Wert kodiert ist.

Es DÜRFEN die folgenden Encodings verwendet werden:

Name	Bedeutung (Encoding)
/nil	Die Zeichenkette ist unkodiert und als Binärdaten zu interpretieren.
/win	Die Zeichenkette ist mittels PDF WinAnsiEncoding (8bit) codiert. (siehe PDF Reference 1.4 [2], Appendix D).
/url	Die Zeichenkette wurde zuerst URL-kodiert (URL-encoded) danach mittels PDF WinAnsiEncoding (8bit) codiert. (siehe PDF Reference 1.4 [2], Appendix D) kodiert.
/f16	Die Zeichenkette ist durch einen 16bit Font dargestellt und entsprechend kodiert. Die Zeichenkette MUSS unter Anwendung der jeweiligen Font-Information wiederhergestellt werden. Dazu MUSS das jeweilige ToUnicode CharacterMapping des verwendeten Fonts im PDF-Dokument eingebettet sein. Aus Gründen der Vereinfachung SOLL der angewendete Font-Zeichensatz auf die im europäischen Sprachraum üblichen Zeichen eingeschränkt sein.

Unbekannte Encoding-Arten MÜSSEN im Zuge der Verifikation der Signatur als nicht spezifikationskonform zurückgewiesen werden.

Wird eine Zeichenkette (Signaturattribut) aus Platzgründen auf mehrere Aussparungen aufgeteilt (siehe Abschnitt 4.3.4.1.3), so MUSS für alle Fragmente dieser Zeichenkette das für das erste Fragment festgelegte Encoding (festgelegt durch das zur entsprechenden Aussparung korrespondierende Element des /encoding-Feldes) angenommen werden. Sind für die anderen Fragmente einer Zeichenkette davon abweichende Encodings festgelegt (festgelegt durch das zur entsprechenden Aussparung korrespondierende Element des /encoding-Feldes), so MÜSSEN diese ignoriert werden.

Das nachfolgende Beispiel zeigt ein exemplarisches /encoding-Feld, im Einklang mit den zuvor, im Beispiel des Abschnitts 4.3.4.1.3, definierten Aussparungen.

Beispiel:

```
[/nil /f16 /nil /nil /win /win /url]
```

4.3.4.1.5 /Cert

Dieses Element wurde in Synergie mit dem /Cert Element von Adobe PDF Signaturen definiert (siehe PDF Reference 1.6 [3], Tabelle 8.98). Es gilt die Einschränkung, dass, falls vorhanden, der Wert immer ein Feld von direkten Literal-String-Objekten sein MUSS. Jeder String MUSS ein Base64-codiertes Zertifikat im PEM Format enthalten. Das erste Zertifikat MUSS das Signaturzertifikat sein. Die restlichen Zertifikate stellen die Zertifikatskette zu einem vertrauenswürdigen Wurzelzertifikat dar und sind OPTIONAL.

4.3.4.1.6 Zeichenketten (Strings) als Wert

Layoutbedingt wird eine längere Zeichenkette der Signatur-Repräsentation auf mehrere Aussparungen aufgeteilt. Umgekehrt muss in der visuellen Darstellung der Signatur-Repräsentation (Signaturblock) im PDF selbst genügend Raum geschaffen werden, um den gesamte Zeichenkette aufnehmen zu können. Dies bedeutet, dass der theoretisch mögliche Platz in Aussparungen nicht voll ausgefüllt werden kann, da diese Zeichenkette visuell, im gewählten Layout des Signaturblocks keinen Platz mehr findet (Darstellung würde bspw. über den Papierrand reichen).

Um die in einer Aussparung gefassten Zeichen/Werte vom ungenutzten Raum der Aussparungen unterscheiden zu können, MÜSSEN alle nicht genutzten Bereiche der Aussparung mit dem NULL-Byte (numerisch 0) befüllt werden. Das NULL-Byte DARF NICHT in einzusetzenden Werten vorkommen.

Vor dem Einsetzen der Werte (kodierte Zeichenketten) in die Bereiche der Aussparungen MÜSSEN, unabhängig der verwendeten und im Feld `/encoding` festgelegten Kodierung, allfällige im einzusetzenden Wert (Zeichenkette) PDF-Steuerzeichen maskiert werden.

Die zu maskierenden PDF-Steuerzeichen sind der Backslash („\“) sowie die linke und rechte Rundklammer („(„ und „)“). Es MUSS jedes im einzusetzenden Wert (Zeichenkette) vorkommende PDF-Steuerzeichen durch einen Backslash eingeleitet (escaped) werden. Es MUSS daher vor dem Einsetzen der Zeichenkette/Werte folgende Ersetzung vorgenommen werden:

1. „\“ werden durch „\\“ ersetzt.
2. „(“ werden durch „\(„ ersetzt.
3. „)“ werden durch „\)“ ersetzt.

Bei der Rekonstruktion der Werte im Zuge der Verifikation MUSS diese Transformation in umgekehrter Reihenfolge durchgeführt und somit rückgängig gemacht werden.

Hinweis: Es MUSS weiters darauf geachtet werden, dass diese ein derart ersetztes PDF-Steuerzeichen – die sogenannte „escape-sequence“ – niemals geteilt wird. Ein alleine stehender „\“ vor dem Ende des Bereichs einer Aussparung würde den Abschluss des Bereichs der Aussparung (End-Delimiter, im PDF mit „)“ dargestellt) „maskieren“ und damit das Dokument unbrauchbar machen.

4.3.5 Anwendungshinweis zur Verifikation

Zur Verifikation von derart signierten Dokumenten MUSS reziprok zu der in dieser Spezifikation festgelegten Vorgehensweise verfahren werden. Zusätzlich werden die folgenden Anwendungshinweise gegeben.

Gegeben sei ein unter Anwendung der hier spezifizierten Signaturmethode textuell signiertes PDF-Dokument. Die Applikation MUSS aus der in der Signatur-Repräsentation enthaltenen Methoden-Kennung das korrekte Signaturverfahren bestimmen und somit das adäquate Verifikationsverfahren anwenden.

Die Vorgehensweise der Verifikation im Überblick:

1. Aus dem zu verifizierenden PDF-Dokument muss der zur Signatur gehörende Incremental Update Block – beinhaltet das relevante EGIZ-Dictionary – extrahiert werden.
2. Über die im EGIZ-Dictionary eingetragenen Byte-Ranges werden die Aussparungen der binären PDF-Signatur ermittelt.
3. Aus den ermittelten Aussparung werden die Werte extrahiert und gemäß den Vorgaben dieser Art (Methode) von binären PDF-Signatur als Daten zur Rekonstruktion der XML-Signatur interpretiert (d.h. Als Signaturattribute, wie Signaturwert, Signaturzeitpunkt, etc., bzw. als Signaturparameter sowie Kennzeichnung des angewandten Signaturparameter-Profils). Diese Daten werden gem. den Vorgaben dieser Art (Methode) von binären PDF-Signatur behandelt, d.h. zusammengeführt und kodiert, und interpretiert.
4. Der Wertebereich der Aussparungen wird gem. den Vorgaben dieser Art (Methode) von binären PDF-Signatur mit den definierten Füllzeichen (Default-Werten) aufgefüllt werden. Das danach resultierende PDF-Dokument repräsentiert das binär signierte PDF-Dokument; dies entspricht dem signierten Datenstrom.

5. Die dem signierten PDF-Dokument hinterlegte XML-Signatur wird anhand der aus dem EGIZ-Dictionary gewonnenen Daten (siehe vorherigen Schritt) unter Berücksichtigung des jeweiligen Signaturparameter-Profils, bzw. unter Anwendung des damit festgelegten XML-Signaturlayouts, rekonstruiert.
6. Die rekonstruierte XML-Signatur wird verifiziert.

4.4 Binäre Signatur, Version 1.1.0

4.4.1 Charakteristik

Methoden-Kennzeichnung:	`urn:pdfsigfilter:bka.gv.at:binaer:v1.1.0`
Input-Datenstrom:	das zu signierende PDF-Dokument (binärer Datenstrom, application/pdf)
Signierte Datenstrom:	das aufbereitet PDF-Dokument (binärer Datenstrom, application/pdf)
Art der Signatur:	XML-Signatur, Detached Signature
Zulässige Signaturparameter:	keine Einschränkung
Anwendbarkeit:	EMPFOHLEN

4.4.2 Aufbereitung der zu signierenden Daten

Der Aufbereitungsprozess ist aus Sicht des Signaturerstellungsprozesses definiert. Bei der Verifikation ist analog vorzugehen (siehe auch Hinweis in Abschnitt 4.4.5).

Die Binäre Signatur sieht vor, dass das gesamte PDF-Dokument binär signiert wird.

Um Manipulationen an einer Binären Signatur auszuschließen, MUSS das Dokument selbst mit samt der vorbereiteten Signatur-Repräsentation (gemäß Vorgaben aus Abschnitt 3) signiert werden. Lediglich die im Zuge der Signaturerstellung gewonnenen Informationen – Signaturwert, Signaturzeitpunkt, Signator bzw. die Signaturattribute der erstellten XML-Signatur im Allgemeinen sowie Angaben zum Signaturzertifikat – MÜSSEN nach der Signaturprozedur in das signierte und vorbereitete PDF-Dokument eingefügt werden. Im Zuge der Signaturprüfung MÜSSEN die nach der Signaturerstellung eingebetteten Werte wieder durch die zum Signaturzeitpunkt verwendeten „Platzhaltern“ ersetzt werden. Dies entspricht somit wieder dem signierten Dokument.

Das binäre signierte PDF-Dokument MUSS zur Signatur vorbereitet werden; dazu MÜSSEN die folgenden Schritte angewendet werden:

1. Dem PDF-Dokument MUSS die Signatur-Repräsentation (Signaturblock) bereits vor der Signaturerstellung eingebettet werden. Dazu ist gemäß den Vorgaben aus Abschnitt 4.3.4 der Signaturblock erstellt und in das Dokument eingebrachte werden. Anstelle der zu diesem Zeitpunkt noch unbekannten Werte (Signaturattribute wie Signaturwert, Signaturzeitpunkt, Angaben zum Signaturzertifikat, etc.) MÜSSEN durch semantisch wertfreie Füllzeichen ersetzt werden. Als Füllzeichen MUSS das NULL-Byte (numerisch 0) verwendet werden (dies ist das Default Füllzeichen für die in der Signatur-Repräsentation vorgesehenen Wertebereiche zur Fassung der Signaturattribute; siehe dazu auch die Vorgaben aus Abschnitt 4.3.4.1.6). Nach erfolgter Signatur werden gemäß den Vorgaben aus Abschnitt 4.3.4 die Signaturattribute in den durch Füllzeichen vorbereiteten Wertebereiche der Signatur-Repräsentation eingefüllt.

Das so vorbereitete PDF-Dokument wird als binärer Datenstrom (Octet Stream) interpretiert und als Datenstrom für die Signaturerstellung herangezogen. Dieser Datenstrom wird signiert.

Der MIME-Type des zu signierenden Datenstroms MUSS daher im Rahmen der XML-Signatur auf `application/pdf` gesetzt werden. Dementsprechend MUSS in den erstellten XML-Signaturen, sofern diese Angaben zu Eigenschaften des signierten Dokumentes beinhalten (z.B. durch das Element `etsi:SignedDataObjectProperties/ etsi:DataObjectFormat`) enthalten, der MIME-Type mit `application/pdf` angegeben werden.

4.4.3 XML-Signaturformat

Die resultierende Signatur ist eine XML Signatur nach [4]. Die zu signierenden Daten MÜSSEN nach Aufbereitung ohne weitere Veränderung als zu signierenden Daten für die Bildung der XML-Signatur herangezogen werden.

Die zu erstellende XML-Signatur MUSS eine Detached Signature sein (siehe [4]).

Die erstellte XML-Signatur folgt den Vorgaben des österreichischen E-Governments bzw. den Vorgaben für XML-Signaturen aus der Spezifikation der österreichischen Bürgerkarte (siehe [6]).

Beispiel einer XML-Signatur nach diesen Vorgaben (erstellt mit der Bürgerkartensoftware IT-Solution trustDesk basic):

```
<dsig:Signature xmlns:dsig="http://www.w3.org/2000/09/xmldsig#" Id="signature-
1201006039-3118750-12731">
  <dsig:SignedInfo>
    <dsig:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-
20010315"/>
    <dsig:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#ecdsa-
sha1"/>
    <dsig:Reference Id="signed-data-reference-0-1201006039-3118750-13501"
URI="urn:Document">
      <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
      <dsig:DigestValue>HASH-WERT 1. REFERENZ</dsig:DigestValue>
    </dsig:Reference>
    <dsig:Reference Id="etsi-data-reference-0-1201006039-3118750-21710"
Type="http://uri.etsi.org/01903/v1.1.1#SignedProperties"
URI="#xmlns(etsi=http://uri.etsi.org/01903/v1.1.1%23)%20xpointer(id('etsi-data-object-
0-1201006039-3118750-
21127')/child::etsi:QualifyingProperties/child::etsi:SignedProperties)">
      <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
      <dsig:DigestValue>HASH-WERT 2. REFERENZ</dsig:DigestValue>
    </dsig:Reference>
  </dsig:SignedInfo>
  <dsig:SignatureValue>SIGNATURWERT</dsig:SignatureValue>
  <dsig:KeyInfo>
    <dsig:X509Data>
      <dsig:X509Certificate>ZERTIFIKAT</dsig:X509Certificate>
    </dsig:X509Data>
  </dsig:KeyInfo>
  <dsig:Object Id="etsi-data-object-0-1201006039-3118750-21127">
    <etsi:QualifyingProperties xmlns:etsi="http://uri.etsi.org/01903/v1.1.1#"
Target="#signature-1201006039-3118750-12731">
      <etsi:SignedProperties>
        <etsi:SignedSignatureProperties>
          <etsi:SigningTime>SIGNATURZEITPUNKT</etsi:SigningTime>
          <etsi:SigningCertificate>
            <etsi:Cert>
              <etsi:CertDigest>
                <etsi:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
                <etsi:DigestValue>HASH-WERT DES ZERTIFIKATS</etsi:DigestValue>
              </etsi:CertDigest>
```

```
          <etsi:IssuerSerial>
            <dsig:X509IssuerName>AUSSTELLER DES ZERTIFIKATS</dsig:X509IssuerName>
            <dsig:X509SerialNumber>SERIENNUMMER DES ZERTIFIKATS</dsig:X509SerialNumber>
          </etsi:IssuerSerial>
        </etsi:Cert>
      </etsi:SigningCertificate>
      <etsi:SignaturePolicyIdentifier>
        <etsi:SignaturePolicyImplied/>
      </etsi:SignaturePolicyIdentifier>
    </etsi:SignedSignatureProperties>
    <etsi:SignedDataObjectProperties>
      <etsi:DataObjectFormat ObjectReference="#signed-data-reference-0-1201006039-
3118750-13501">
        <etsi:MimeType>application/pdf</etsi:MimeType>
      </etsi:DataObjectFormat>
    </etsi:SignedDataObjectProperties>
  </etsi:SignedProperties>
 </etsi:QualifyingProperties>
</dsig:Object>
</dsig:Signature>
```

Dieses Beispiel enthält einige Besonderheiten der Signaturerstellungskomponente (Bürgerkartensoftware und Signaturerstellungseinheit), auf die in Verbindung mit Signaturparametern noch eingegangen wird. Aus Gründen der Übersichtlichkeit wurden variable Inhalte größtenteils durch verbale Umschreibungen ersetzt (eingerahmter Text).

4.4.4 Einbettung der Signatur in das PDF-Dokument

Die resultierende XML-Signatur MUSS letztlich in Form der in Abschnitt 3 definierten Repräsentation in das PDF-Dokument integriert werden.

Die Einbettung der Signatur-Repräsentation (Signaturblock) im PDF-Dokument MUSS mit Hilfe eines Inkrementellen Update Blocks (Incremental Update Block, Abschnitt 3.4.5 in [2]) realisiert werden. Dieser Block MUSS folgende Struktur aufweisen:

1. Dieser Incremental Update Block MUSS ein eigenes Dictionary, das EGIZ-Dictionary, enthalten. Dieses MUSS ein indirektes Objekt sein.
2. Der gesamte sichtbare Signaturblock MUSS in ein XObject Form eingebettet sein. (siehe /SigXObject Key des EGIZ Dictionaries, Abschnitt 4.4.4.1)
3. Das trailer-Dictionary des Incremental Update Blocks MUSS einen Key /EGIZSigDict enthalten. Wert dieses Keys MUSS eine indirekte Referenz auf das EGIZ-Dictionary sein.

Der nachfolgende Abschnitt beschreibt das EGIZ-Dictionary im Detail.

Als Spezifikum dieses Algorithmus MUSS die Signatur-Repräsentation bereits vor dem Signaturvorgang, im Zuge der Aufbereitung der zu signierenden Daten, in das PDF-Dokument eingebracht werden. Anstelle der zu diesem Zeitpunkt noch unbekannten Werte, wie bspw. Signaturattribute (das sind zum Beispiel Signaturwert, Signaturzeitpunkt, Angaben zum Signaturzertifikat, etc.), MÜSSEN die dafür vorgesehenen Wertebereiche mit semantisch wertfreien Füllzeichen aufgefüllt werden.

Das mit diesem vorbereiteten aber leeren Signaturblock versehene PDF-Dokument wird in seiner binären Repräsentation elektronisch signiert. Nach dem Signaturprozess MÜSSEN die dabei ermittelten Werte (Signaturattribute) in die dafür vorgesehenen Wertebereiche der Signatur-Repräsentation (in den nachfolgenden Abschnitten auch als ‚Aussparung' bezeichnet) eingefüllt werden.

4.4.4.1 EGIZ-Dictionary

Die Keys des EGIZ Dictionaries MÜSSEN, falls verwendet, in folgender Reihenfolge vorhanden sein:

Key	Kardinalität	Beschreibung
/Type	MUSS	Bezeichnet den Typ des Dictionaries. Es MUSS /EGIZSigDict lauten.
/ODS	MUSS	Enthält die Größe des gesamten Dokuments (Originaldokument inklusive Incremental Update Block für die Signatur-Repräsentation). Alle Byte-Ranges MÜSSEN innerhalb dieses Werts liegen.
/ID	MUSS	Enthält die Byte-Range der Zeichenkette (String), die die angewendete Signaturmethode (Methode) identifiziert. Siehe Abschnitt 4.4.4.1.1.
/SigXObject	MUSS	Enthält die indirekte Referenz auf das XObject Form der Signatur-Repräsentation. Diese MUSS eine indirekte Referenz sein.
/ByteRange	MUSS	Enthält aufsteigend sortierte Byte-Ranges, wodurch die signierten Bereiche des Dokumentes auf binärer Ebene identifiziert werden. Dieser Array ist analog zu den Byte-Range Arrays der Adobe PDF Signaturen definiert (siehe PDF Reference 1.6 [3], Kapitel 8.7, Tabelle 8.98). Die erste Byte-Range MUSS an Position 0 beginnen und die letzte Byte-Range dieses Arrays MUSS das Ende des Dokumentes referenzieren. Siehe Abschnitt 4.4.4.1.2.
/replaces	MUSS	Dieses Feld gibt an, welche Elemente der Signatur-Repräsentation nach der Signaturerstellung durch signaturspezifische Werte ersetzt werden müssen. Das Feld stellt ein PDF Array von PDF-Names dar, welche angeben, welches Signaturattribut in den Aussparungen der binären Signatur – binär repräsentiert durch die inversen Byte-Ranges des /ByteRange-Elements – enthalten sind. Gibt es in einem Dokument n Byte-Ranges, so enthält das /replaces-Array (n-1) Einträge für die (n-1) Bereiche. Siehe Abschnitt 4.4.4.1.3.
/encodings	MUSS	Das Feld stellt ein PDF Array von PDF-Names dar, welche angeben, welches Encoding bei den Daten der Signaturattribute angewendet wurde. Die Reihenfolge der in diesem Array angegebenen Encodings MUSS mit der Reihenfolge des /replaces-Arrays korrespondieren. Siehe Abschnitt 4.4.4.1.4.

Key	Kardinalität	Beschreibung
/Cert	SOLL	Dieses Array von PDF-Strings KANN zur Einbettung von Zertifikatsdaten (X509-Zertifiakten) verwendet werden. Es SOLL zumindest das Signaturzertifikat selbst, es KANN jedoch auch die gesamte Zertifikatskette im Rahmen dieses Arrays als PDF-Strings eingebettet werden. Siehe Abschnitt 4.4.4.1.5.

Sämtliche Werte, falls nicht anders spezifiziert, MÜSSEN direkte Objekte, also direkt im EGIZ-Dictionary eingebettet sein.

Das EGIZ-Dictionary DARF noch weitere Elemente enthalten. Diese können dazu verwendet werden die Signatur mit zusätzlicher Information auszustatten.

4.4.4.1.1 /ID

Das /ID-Element beschreibt die Byte-Ranges (siehe Beschreibung /ByteRange) der Zeichenkette (String), die die angewendete Signaturmethode (Methode) identifiziert.

Der so identifizierte String ist jener Teil des Content Streams der Signatur-Repräsentation, welcher den Text der Signaturmethode enthält. Dieser String unterliegt genauso den PDF Formatierungsregeln und wird daher im Allgemeinen in mehrere PDF-Strings gebrochen sein.

Die /ID Byte-Ranges beschreiben den Inhalt dieser PDF-Strings des Content Streams.

Zusammengefügt MÜSSEN die durch diese Byte-Ranges spezifizierten Strings wieder den Signaturmethoden-String ergeben. Das Character Encoding des Kennzeichnungsstrings MUSS 8 bit WinAnsiEncoding sein. Zu beachten sind auch PDF String Escape Sequences (siehe Kapitel „Strings in den Löchern“).

4.4.4.1.2 /ByteRange

Statische Bereiche werden durch Byte-Ranges beschrieben. Byte-Ranges sind in Analogie zu Byte-Ranges in Adobe PDF-Signaturen definiert (siehe PDF Reference 1.6 [3], Kapitel 8.7):

- Eine Byte-Range MUSS aus dem Zahlentupel Startoffset und Länge (in Bytes) bestehen.
- Beide MÜSSEN positive Ganzzahlen sein, wobei der Startoffset auch 0 sein DARF.
- Der Startoffset MUSS vom Anfang der PDF-Datei an gemessen werden.
- Die Länge MUSS die Anzahl an Bytes ab dem Startoffset angeben, welche zur Byte-Range gehören.

In einer binären Signatur MÜSSEN alle statischen Bereiche mittels Byte-Ranges identifiziert werden. Die variablen Bereiche zwischen den Byte-Ranges werden als Aussparungen bezeichnet, da die dadurch binär identifizierten Bereiche des PDF-Dokumentes nicht von der Signatur abgedeckt werden. In die Bereiche dieser Aussparungen MÜSSEN zum Signierzeitpunkt sogenannte Platzhalter-Zeichen eingefügt sein. Nach erfolgter Signatur MÜSSEN diese durch die resultierenden Werte der Signatur (zum Beispiel Signaturwert, Signaturzeitpunkt, etc.) ersetzt werden.

Die Angaben von Byte-Ranges in den Elementen der binären Signatur MÜSSEN immer gemäß ihrer Startoffsets aufsteigend sortiert sein.

Beispiel:

Die Byte-Range (10, 5) beschreibt die Bytes an den Positionen 10, 11, 12, 13 und 14. Ein korrespondierendes `/ByteRange`-Array könnte folgendermaßen aussehen:

```
[ 0 100 110 90 ]
```

Dieses beschreibt zwei Byte-Ranges – von 0 bis 99 sowie von 110 bis 199 – und eine Aussparung – von 100 bis 109.

Mit Hilfe dieser Byte-Range Angaben werden die signierten Bereiche des Dokuments auf Byte-Ebene identifiziert. Umgekehrt werden damit jene, nichtsignierten Aussparungen identifiziert, in die nach der Signatur die Signaturattribute eingebettet werden müssen. Das nachfolgende Beispiel soll dies illustrieren.

Beispiel:

```
[…]
1 0 0 1 137.91 207 Tm
/F1 12 Tf
(123456789a)Tj
ET
BT
1 0 0 1 217.11 225 Tm
/F2 12 Tf
(123bcdefgh)Tj
1 0 0 1 217.11 213 Tm
[…]
```

Die nicht umrahmten Bereiche sind jene Bereiche, die über die Angabe von Byte-Ranges als die „signierten Bytes“ des Dokuments identifiziert werden würden. Die umrahmten Bereiche stellen hingegen Aussparungen dar, welche im signierten Dokument durch semantisch wertfreie Platzhalter ersetzt werden. Diese Struktur wird durch das `/ByteRange`-Element auf Byte-Ebene beschrieben.

4.4.4.1.3 /replaces

Das Feld stellt ein PDF Array von PDF-Names dar, welche angeben, welches Signaturattribut in den Aussparungen der binären Signatur – binär repräsentiert durch die inversen Byte-Ranges des `/ByteRange`-Elements – enthalten sind.

Es MUSS so viele Elemente im `/replaces` Array geben wie es Aussparungen durch die getroffene Definition der Byte-Ranges gibt. Gibt es in einem Dokument n Byte-Ranges, so enthält das `/replaces`-Array (n-1) Einträge für die (n-1) Aussparungen.

Jedes Element im `/replaces` Array MUSS ein gültiger PDF-Name sein, welcher den semantischen Wert (Typ) des Inhalts der Aussparung festlegt.

Folgende PDF-Namen sind zur Beschreibung der semantischen Werte definiert:

PDF-Name	Semantischer Wert (Typ) / Signaturattribut
`/nil`	Keine für die Signatur relevante Information oder der Typ des Wertes wird auf andere Art und Weise bestimmt, zum Beispiel durch eine explizite Referenz im EGIZ-Dictionary (siehe Definition des `/Cert`-Array).
`/dat`	Zeichenkette, die den Signaturzeitpunkt repräsentiert/beschreibt (oder Teilfragment dessen).

PDF-Name	Semantischer Wert (Typ) / Signaturattribut
/iss	Zeichenkette, die den Aussteller des Signaturzertifikates repräsentiert/beschreibt (oder Teilfragment dessen).
/snr	Zeichenkette, die den Seriennummer des Signaturzertifikates repräsentiert/beschreibt (oder Teilfragment dessen).
/val	Zeichenkette, die den Signaturwert repräsentiert/beschreibt (oder Teilfragment dessen).
/sid	Zeichenkette, die die Signaturparameter repräsentiert/beschreibt (oder Teilfragment dessen).

Unbekannte Typen MÜSSEN im Zuge der Verifikation der Signatur als nicht Spezifikationskonform zurückgewiesen werden.

Signaturattribute, bzw. die sie repräsentierenden Zeichenketten, DÜRFEN aus Platzgründen auf mehrere, aufeinanderfolgende Aussparungen aufgeteilt werden. Der Inhalt von derart aufeinander folgenden Aussparungen gleichen Typs stellt zusammengefasst den Gesamtwert des betreffenden Signaturattributs dar. Um diesen Gesamtwert zu bilden MÜSSEN die Fragmente konkateniert werden.

Das folgende Beispiel zeigt die Beschreibung der semantischen Inhalte von Aussparungen, wobei die erste Aussparung ein nicht näher spezifizierte Datenblock ist (Typ `/nil`), die zweite und die dritte Aussparung Fragmente der Zeichenkette des Signaturdatums enthalten (Typ `/dat`), die Aussparung vier bis sechs Fragmente der Zeichenkette des Signaturwertes enthalten (Typ `/val`), die siebte und achte Aussparung wiederum nicht näher spezifizierte Datenblöcke enthalten, die letzte Aussparung die die Signaturparameter beschreibende Zeichenkette repräsentiert (Typ `/sid`).

Beispiel:

```
[/nil /dat /dat /val /val /val /nil /nil /sid]
```

4.4.4.1.4 /encodings

Die Festlegung der Typen der in Aussparungen gefassten Werte durch das /replaces-Feld werden durch die analoge Festlegung von Kodierungsarten (Encoding) ergänzt. Das /encodings-Feld legt daher fest, auf welche Art die in den jeweiligen Aussparungen gefassten Daten (vor allem Zeichenketten) kodiert worden sind.

Es MUSS für jeden Wert, der in einer Aussparung nach der Signatur eingebettet und im /replaces-Feld entsprechend festgelegt wurde, spezifiziert werden, auf welche Art dieser Wert kodiert ist.

Es DÜRFEN die folgenden Encodings verwendet werden:

Name	Bedeutung (Encoding)
/nil	Die Zeichenkette ist unkodiert und als Binärdaten zu interpretieren.
/win	Die Zeichenkette ist mittels PDF `WinAnsiEncoding` (8bit) codiert. (siehe PDF Reference 1.4 [2], Appendix D).
/url	Die Zeichenkette wurde zuerst URL-kodiert (URL-encoded) danach mittels PDF `WinAnsiEncoding` (8bit) codiert. (siehe PDF Reference 1.4 [2], Appendix D) kodiert.

Name	Bedeutung (Encoding)
/f16	Die Zeichenkette ist durch einen 16bit Font dargestellt und entsprechend kodiert. Die Zeichenkette MUSS unter Anwendung der jeweiligen Font-Information wiederhergestellt werden. Dazu MUSS das jeweilige ToUnicode CharacterMapping des verwendeten Fonts im PDF-Dokument eingebettet sein. Aus Gründen der Vereinfachung SOLL der angewendete Font-Zeichensatz auf die im europäischen Sprachraum üblichen Zeichen eingeschränkt sein.

Unbekannte Encoding-Arten MÜSSEN im Zuge der Verifikation der Signatur als nicht spezifikationskonform zurückgewiesen werden.

Wird eine Zeichenkette (Signaturattribut) aus Platzgründen auf mehrere Aussparungen aufgeteilt (siehe Abschnitt 4.3.4.1.3), so MUSS für alle Fragmente dieser Zeichenkette das für das erste Fragment festgelegte Encoding (festgelegt durch das zur entsprechenden Aussparung korrespondierende Element des /encoding-Feldes) angenommen werden. Sind für die anderen Fragmente einer Zeichenkette davon abweichende Encodings festgelegt (festgelegt durch das zur entsprechenden Aussparung korrespondierende Element des /encoding-Feldes), so MÜSSEN diese ignoriert werden.

Das nachfolgende Beispiel zeigt ein exemplarisches /encoding-Feld, im Einklang mit den zuvor, im Beispiel des Abschnitts 4.3.4.1.3, definierten Aussparungen.

Beispiel:

```
[/nil /f16 /nil /nil /win /win /url]
```

4.4.4.1.5 /Cert

Dieses Element wurde in Synergie mit dem `/Cert` Element von Adobe PDF Signaturen definiert (siehe PDF Reference 1.6 [3], Tabelle 8.98). Es gilt die Einschränkung, dass, falls vorhanden, der Wert immer ein Feld von direkten Literal-String-Objekten sein MUSS. Jeder String MUSS ein Base64-codiertes Zertifikat im PEM Format enthalten. Das erste Zertifikat MUSS das Signaturzertifikat sein. Die restlichen Zertifikate stellen die Zertifikatskette zu einem vertrauenswürdigen Wurzelzertifikat dar und sind OPTIONAL.

4.4.4.1.6 Zeichenketten (Strings) als Wert

Layoutbedingt wird eine längere Zeichenkette der Signatur-Repräsentation auf mehrere Aussparungen aufgeteilt. Umgekehrt muss in der visuellen Darstellung der Signatur-Repräsentation (Signaturblock) im PDF selbst genügend Raum geschaffen werden, um den gesamte Zeichenkette aufnehmen zu können. Dies bedeutet, dass der theoretisch mögliche Platz in Aussparungen nicht voll ausgefüllt werden kann, da diese Zeichenkette visuell, im gewählten Layout des Signaturblocks keinen Platz mehr findet (Darstellung würde bspw. Über den Papierrand reichen).

Um die in einer Aussparung gefassten Zeichen/Werte vom ungenutzten Raum der Aussparungen unterscheiden zu können, MÜSSEN alle nicht genutzten Bereiche der Aussparung mit dem NULL-Byte (numerisch 0) befüllt werden. Das NULL-Byte DARF NICHT in einzusetzenden Werten vorkommen.

Vor dem Einsetzen der Werte (kodierte Zeichenketten) in die Bereiche der Aussparungen MÜSSEN, unabhängig der verwendeten und im Feld `/encoding` festgelegten Kodierung, allfällige im einzusetzenden Wert (Zeichenkette) PDF-Steuerzeichen maskiert werden.

Die zu maskierenden PDF-Steuerzeichen sind der Backslash („\“) sowie die linke und rechte Rundklammer („(„ und „)“). Es MUSS jedes im einzusetzenden Wert (Zeichenkette) vorkommende PDF-Steuerzeichen durch einen Backslash eingeleitet (escaped) werden. Es MUSS daher vor dem Einsetzen der Zeichenkette/Werte folgende Ersetzung vorgenommen werden:

1. „\“ werden durch „\\“ ersetzt.
2. „(“ werden durch „\(„ ersetzt.
3. „)“ werden durch „\)“ ersetzt.

Bei der Rekonstruktion der Werte im Zuge der Verifikation MUSS diese Transformation in umgekehrter Reihenfolge durchgeführt und somit rückgängig gemacht werden.

Hinweis: Es MUSS weiters darauf geachtet werden, dass diese ein derart ersetztes PDF-Steuerzeichen – die sogenannte „escape-sequence“ – niemals geteilt wird. Ein alleine stehender „\“ vor dem Ende des Bereichs einer Aussparung würde den Abschluss des Bereichs der Aussparung (End-Delimiter, im PDF mit „)“ dargestellt) „maskieren“ und damit das Dokument unbrauchbar machen.

4.4.5 Anwendungshinweis zur Verifikation

Zur Verifikation von derart signierten Dokumenten MUSS reziprok zu der in dieser Spezifikation festgelegten Vorgehensweise verfahren werden. Zusätzlich werden die folgenden Anwendungshinweise gegeben.

Gegeben sei ein unter Anwendung der hier spezifizierten Signaturmethode textuell signiertes PDF-Dokument. Die Applikation MUSS aus der in der Signatur-Repräsentation enthaltenen Methoden-Kennung das korrekte Signaturverfahren bestimmen und somit das adäquate Verifikationsverfahren anwenden.

Die Vorgehensweise der Verifikation im Überblick:

1. Aus dem zu verifizierenden PDF-Dokument muss der zur Signatur gehörende Incremental Update Block – beinhaltet das relevante EGIZ-Dictionary – extrahiert werden.
2. Über die im EGIZ-Dictionary eingetragenen Byte-Ranges werden die Aussparungen der binären PDF-Signatur ermittelt.
3. Aus den ermittelten Aussparung werden die Werte extrahiert und gemäß den Vorgaben dieser Art (Methode) von binären PDF-Signatur als Daten zur Rekonstruktion der XML-Signatur interpretiert (d.h. Als Signaturattribute, wie Signaturwert, Signaturzeitpunkt, etc., bzw. als Signaturparameter sowie Kennzeichnung des angewandten Signaturparameter-Profils). Diese Daten werden gem. den Vorgaben dieser Art (Methode) von binären PDF-Signatur behandelt, d.h. zusammengeführt und kodiert, und interpretiert.
4. Der Wertebereich der Aussparungen wird gem. den Vorgaben dieser Art (Methode) von binären PDF-Signatur mit den definierten Füllzeichen (Default-Werten) aufgefüllt werden. Das danach resultierende PDF-Dokument repräsentiert das binär signierte PDF-Dokument; dies entspricht dem signierten Datenstrom.
5. Die dem signierten PDF-Dokument hinterlegte XML-Signatur wird anhand der aus dem EGIZ-Dictionary gewonnenen Daten (siehe vorherigen Schritt) unter Berücksichtigung des jeweiligen Signaturparameter-Profils, bzw. unter Anwendung des damit festgelegten XML-Signaturlayouts, rekonstruiert.
6. Die rekonstruierte XML-Signatur wird verifiziert.

5 Definierte Signaturparameter

Die Signaturparameter nehmen auf Spezialitäten von Signaturerstellungskomponenten Rücksicht. Vor allem werden damit das Layout und die variablen Elemente der damit erstellten XML-Signaturen im Rahmen der Signatur-Repräsentation (Signaturblock) zum Ausdruck gebracht. Dies ist besonders für den Fall der Signaturrekonstruktion auf Basis eines Papierausdruckes erforderlich (siehe auch allgemeine Erläuterung in Abschnitt 2.2).

Die vorliegende Spezifikation berücksichtigt Spezialitäten einiger Signaturerstellungs-komponenten und enthält daher die Definition der folgenden Signaturparameter-Profile, die wie folgt in Implementierungen unterstützt werden MÜSSEN:

Signaturparameter	Status	In Implementierungen zu unterstützen bei	
		Verifikation	Signatur
Default (MOA)	EMPFOHLEN	MUSS	MUSS
Default (BKU)	DEPRECATED	EMPFOHLEN	NICHT EMPFOHLEN
`etsi-bka-1.0`	EMPFOHLEN	MUSS	MUSS
`etsi-moc-1.0`	EMPFOHLEN	MUSS	MUSS

Eine spezifikationskonforme Umsetzung MUSS die in der obigen Tabelle definierten Signaturparameterprofile, gemäß den definierten Prioritäten, implementieren.

Jede Implementierung MUSS für die damit erzeugbaren Signaturparameterprofile sowohl die Signaturerstellung als auch die Signaturverifikation realisieren.

5.1 Default Signaturparameter-Profil

5.1.1 Charakteristik

Parameter-Kennzeichnung: keine Kennzeichnung; es werden keine Parameter angeführt

Signaturerstellungskomponente: für alle Signaturerstellungskomponenten gem. Spezifikation MOA-SS [7]

Einschränkungen bzgl. Signatur-methoden: keine, kann mit allen Signaturmethoden verwendet werden

Anwendbarkeit: EMPFOHLEN

5.1.2 Signaturparameter

Für dieses Signaturparameter-Profil DÜRFEN Signaturparameter NICHT verwendet werden.

5.1.3 Signaturlayout

Werden keine Signaturparameter angegeben, so MUSS die erstellte Signatur den Vorgaben einer MOA-SS Signatur [7] folgen und DARF NICHT variable (zeitabhängige oder zufällige) Werte enthalten.

Eine XML-Signatur, die diesem Parameter-Profil entspricht, MUSS dem folgenden Layout entsprechen:

```
<dsig:Signature Id="signature-1-1" xmlns:dsig="http://www.w3.org/2000/09/xmldsig#">
 <dsig:SignedInfo xmlns:dsig="http://www.w3.org/2000/09/xmldsig#">
  <dsig:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-
20010315"/>
  <dsig:SignatureMethod Algorithm="ALGORITHM"/>
  <dsig:Reference Id="reference-1-1" URI="REFERENCE">
   TRANSFORMS
    <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
    <dsig:DigestValue>DIGESTVALUESIGNEDDATA</dsig:DigestValue>
  </dsig:Reference>
  <dsig:Reference Type="http://uri.etsi.org/01903/v1.1.1#SignedProperties"
URI="#xmlns(etsi=http://uri.etsi.org/01903/v1.1.1%23)%20xpointer(id('etsi-signed-1-
1')/child::etsi:QualifyingProperties/child::etsi:SignedProperties)">
    <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
    <dsig:DigestValue>DIGESTVALUESIGNEDPROPERTIES</dsig:DigestValue>
  </dsig:Reference>
 </dsig:SignedInfo>
 <dsig:SignatureValue>SIGNATUREVALUE</dsig:SignatureValue>
 <dsig:KeyInfo>
  <dsig:X509Data>
   <dsig:X509Certificate>X509CERTIFICATE</dsig:X509Certificate>
  </dsig:X509Data>
 </dsig:KeyInfo>
 DSIGOBJECT
 <dsig:Object Id="etsi-signed-1-1">
  <etsi:QualifyingProperties Target="#signature-1-1"
xmlns:etsi="http://uri.etsi.org/01903/v1.1.1#">
   <etsi:SignedProperties xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
xmlns:etsi="http://uri.etsi.org/01903/v1.1.1#">
    <etsi:SignedSignatureProperties>
     <etsi:SigningTime>SIGNINGTIME</etsi:SigningTime>
     <etsi:SigningCertificate>
      <etsi:Cert>
       <etsi:CertDigest>
        <etsi:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
        <etsi:DigestValue>DIGESTVALUEX509CERTIFICATE</etsi:DigestValue>
       </etsi:CertDigest>
       <etsi:IssuerSerial>
        <dsig:X509IssuerName>X509ISSUERNAME</dsig:X509IssuerName>
        <dsig:X509SerialNumber>X509SERIALNUMBER</dsig:X509SerialNumber>
       </etsi:IssuerSerial>
      </etsi:Cert>
     </etsi:SigningCertificate>
     <etsi:SignaturePolicyIdentifier>
      <etsi:SignaturePolicyImplied/>
     </etsi:SignaturePolicyIdentifier>
    </etsi:SignedSignatureProperties>
```

```
      <etsi:SignedDataObjectProperties>
        <etsi:DataObjectFormat ObjectReference="#reference-1-1">
          <etsi:MimeType>MIMETYPE</etsi:MimeType>
        </etsi:DataObjectFormat>
      </etsi:SignedDataObjectProperties>
    </etsi:SignedProperties>
  </etsi:QualifyingProperties>
 </dsig:Object>
</dsig:Signature>
```

Die vom Layout vorgesehenen Variabilitäten werden in Großbuchstaben und umrandet ausgewiesen. Die nachfolgenden Abschnitte definieren diese im Detail.

Dieses Profil legt durch das damit festgelegte XML-Signaturlayout die folgenden wesentlichen XML-Signatur-Eigenschaften implizit fest:

1. Kanonisierungsmethode:
   `http://www.w3.org/TR/2001/REC-xml-c14n-20010315`
2. Digest-Methode: `http://www.w3.org/2000/09/xmldsig#sha1`
3. Qualifying Properties: nach ETSI (`http://uri.etsi.org/01903/v1.1.1#`)

5.1.3.1 ALGORITHM

Legt den Signaturalgorithmus fest. Es MUSS einer der folgenden Signaturalgorithmen verwendet werden, in Abhängigkeit der Eigenschaften des Signatur-Schlüssels:

- für ECDSA-Schlüssel: `http://www.w3.org/2000/09/xmldsig#rsa-sha1`
- für RSA-Schlüssel: `http://www.w3.org/2000/09/xmldsig#rsa-sha1`

Der zum Signaturschlüssel passende Identifikator (URI) MUSS im XML-Signaturlayout anstelle der Variable **ALGORITHM** eingesetzt werden.

5.1.3.2 REFERENCE

Wird eine Signatur-Methode angewandt, die eine enveloping XML-Signatur erzeugt, so MUSS das folgende Fragment anstelle der Variable **REFERENCE** im XML-Signaturlayout eingesetzt werden:

```
#xmlns(etsi=http://uri.etsi.org/01903/v1.1.1%23)%20xpointer(id('ets
i-signed-1-1')/child::etsi:QualifyingProperties/child::etsi:Sig
nedProperties)
```

Wird eine Signatur-Methode angewandt, die eine detached XML-Signatur erzeugt, so MUSS das folgende Fragment anstelle der Variable **REFERENCE** im XML-Signaturlayout eingesetzt werden:

```
urn:Document
```

5.1.3.3 TRANSFORMS

Wird eine Signatur-Methode angewandt, die eine enveloping XML-Signatur erzeugt, so MUSS das folgende Fragment anstelle der Variable **TRANSFORMS** im XML-Signaturlayout eingesetzt werden:

```
<dsig:Transforms>
  <dsig:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#base64"/>
</dsig:Transforms>
```

Wird eine Signatur-Methode angewandt, die eine detached XML-Signatur erwirkt, so darf kein Transformationspfad eingefügt werden. In diesem Fall DARF ein Transformationspfad NICHT im XML-Signaturlayout eingesetzt werden. Die Variable **TRANSFORMS** im XML-Signaturlayout MUSS ersatzlos entfernt werden.

5.1.3.4 DIGESTVALUESIGNEDDATA

Anstelle der Variable **DIGESTVALUESIGNEDDATA** MUSS im XML-Signaturlayout der Hash-Wert der ersten XML-Signaturreferenz eingesetzt werden.

5.1.3.5 DIGESTVALUESIGNEDPROPERTIES

Anstelle der Variable **DIGESTVALUESIGNEDPROPERTIES** MUSS im XML-Signaturlayout der Hash-Wert der zweiten XML-Signaturreferenz eingesetzt werden.

5.1.3.6 SIGNATUREVALUE

Anstelle der Variable **SIGNATUREVALUE** MUSS im XML-Signaturlayout der Signaturwert eingesetzt werden.

5.1.3.7 X509CERTIFICATE

Anstelle der Variable **X509CERTIFICATE** MUSS im XML-Signaturlayout das Base64-kodierte Signaturzertifikat (X509-Zertifikat, kodiert gem. Abschnitt 4.4.4 in [4]) eingesetzt werden.

5.1.3.8 DSIGOBJECT

Wird eine Signatur-Methode angewandt, die eine enveloping XML-Signatur erzeugt, so MUSS das folgende Fragment anstelle der Variable **DSIGOBJECT** im XML-Signaturlayout eingesetzt werden:

```
<dsig:Object Id="signed-data-1-1-1">
  <Base64Content>BASE64CONTENT</Base64Content>
</dsig:Object>
```

In diesem Fragment MUSS anstelle der Variable **BASE64CONTENT** der Base64-kodierte, zu signierende Datenstrom (gem. angewandter Signaturmethode) eingesetzt werden.

Wird eine Signatur-Methode angewandt, die eine detached XML-Signatur erzeugt, so ist kein XML-Signaturobjekt für die signierten Daten notwendig. Die Variable **DSIGOBJECT** im XML-Signaturlayout MUSS ersatzlos entfernt werden.

5.1.3.9 SIGNINGTIME

Anstelle der Variable **SIGNINGTIME** MUSS der Signaturzeitpunkt gem. [5] eingesetzt werden.

5.1.3.10 DIGESTVALUEX509CERTIFICATE

Anstelle der Variable **DIGESTVALUEX509CERTIFICATE** MUSS der Hash-Wert des Signaturzertifikates (lt. [5]) eingesetzt werden.

5.1.3.11 X509ISSUERNAME

Anstelle der Variable **X509ISSUERNAME** MUSS der Name des Ausstellers des Signaturzertifikates (lt. [5]) eingesetzt werden.

5.1.3.12 X509SERIALNUMBER

Anstelle der Variable **X509SERIALNUMBER** MUSS die Seriennummer des Signaturzertifikates (lt. [5]) eingesetzt werden.

5.1.3.13 MIMETYPE

Anstelle der Variable **MIMETYPE** MUSS der MIME-Type der zu signierenden Daten eingesetzt werden. Der MIME-Type ist von der angewandten Signaturmethode abhängig und MUSS im Zuge der Definition der Signaturmethode festgelegt werden.

5.2 Default Signaturparameter-Profil für BKU

5.2.1 Charakteristik

Parameter-Kennzeichnung:	keine Kennzeichnung; es werden zwar explizite Signaturparameter eingeführt, dieses Profil hat jedoch keine eigene Kennzeichnung. Es werden nur die Parameter in der Signatur-Repräsentation (Signaturblock) eingefügt.
Signaturerstellungskomponente:	dieses Profil MUSS für die Generallizenz der Bürgerkartenumgebung (IT-Solution trustDesk-Basic), bis zur Version 2.7.4. angewendet werden.
Einschränkungen bzgl. Signatur-methoden:	dieses Profil MUSS mit Signaturmethoden verwendet werden, die eine enveloping Signatur erzeugen; diese sind: `urn:pdfsigfilter:bka.gv.at:binaer:v1.0.0` und `urn:pdfsigfilter:bka.gv.at:text:v1.0.0`
Verwendbarkeit:	Dieses Signaturparameter-Profil DARF NICHT mehr zur Anwendung gelangen (deprecated). Es wurde durch das `ets-bka-1.0` Signaturparameter-Profil ersetzt.

5.2.2 Signaturparameter

Für dieses Signaturparameter-Profil wurden Signaturparameter definiert. Diese MÜSSEN ohne vorangestellte Kennzeichnung in die Signatur-Repräsentation eingefügt werden.

Für die Formulierung der Signaturparameter MUSS die Definition gem. Abschnitt 2.2. Es gilt bei diesem Signaturparameter-Profil jedoch die Maßgabe, dass das Feld `<SIGDEV_PROF>` leer bleiben MUSS.

Die Signaturparameter dieses Profils repräsentieren 5 Einzelwerte. Dabei MUSS als Parameter Teil 1 (Feld `PARAM_L1` der Signaturparameter, siehe 2.2) als konstanter Präfix für alle 5 Einzelwerte herangezogen werden. Der Parameter Teil 2 (Feld `PARAM_L2` der Signaturparameter, siehe 2.2) enthält selbst 5 Einzelwerte, die jeweils mit einem Bindestrich separiert werden müssen. Es gilt für Parameter Teil 2 folgende Ausformung (in Ergänzung zur Definition aus Abschnitt 2.2):

```
<PARAM_L2> :: = <WERT_1>"-"<WERT_2>"-"<WERT_3>"-"<WERT_4>"-"<WERT_5>
   <WERT_1> :: = 1*<CHAR>
   <WERT_2> :: = 1*<CHAR>
   <WERT_3> :: = 1*<CHAR>
   <WERT_4> :: = 1*<CHAR>
   <WERT_5> :: = 1*<CHAR>
```

Daraus MÜSSEN folgende Einzelwerte gebildet werden:

```
<ParamSigID>           ::= <PARAM_L1>"-"<WERT_1>
<ParamSigDataRef>      ::= "0-"<PARAM_L1>"-"<WERT_2>
<ParamSigDataObjURI>   ::= "0-"<PARAM_L1>"-"<WERT_3>
<ParamEtsiDataRef>     ::= "0-"<PARAM_L1>"-"<WERT_4>
<ParamEtsiDataObjURI>  ::= "0-"<PARAM_L1>"-"<WERT_5>
```

Dieser Werte werden mehrfach im Signaturlayout referenziert und verwendet. Wird eine Signatur-Methode angewandt, die eine detached XML-Signatur erzeugt, so wird der Wert `<ParamSigDataObjURI>` nicht benötigt. Die Signaturparameter lassen die Bildung dieses Wertes jedoch dennoch zu.

Die Verwendung dieser Einzelwerte wird in Abschnitt 5.2.3 festgelegt.

Nachfolgend ein Beispiel zur Bildung der Einzelwerte auf Basis übergebener Signaturparameter (spezifisch für das vorliegende Signaturparameter-Profil):

Signaturparameter lt. Signaturrepräsentation:

```
@1200412799-27800484@23524-22018-26789-24095-30271
```

Daraus ergeben sich:

```
<ParamSigID>          = 1200412799-27800484-23524
<ParamSigDataRef>     = 0-1200412799-27800484-22018
<ParamSigDataObjURI>  = 0-1200412799-27800484-26789
<ParamEtsiDataRef>    = 0-1200412799-27800484-24095
<ParamEtsiDataObjURI> = 0-1200412799-27800484-30271
```

5.2.3 Signaturlayout

Die Signaturen einer Bürgerkartenumgebung enthalten zeitabhängige Varianzen. Diese sind vor allem eine Reihe von XML-Attributen (ID-Attribute) die zur Referenzierung von XML-Elementen/-Knoten herangezogen werden. Diese variablen Attribute MÜSSEN in Form der Signaturparameter innerhalb der Signatur-Repräsentation verwaltet und bei der Rekonstruktion der Signatur entsprechend berücksichtigt werden.

Eine XML-Signatur, die diesem Parameter-Profil entspricht, MUSS dem folgenden Layout entsprechen:

```
<dsig:Signature xmlns:dsig="http://www.w3.org/2000/09/xmldsig#" Id="signature-SIGID">
  <dsig:SignedInfo>
    <dsig:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-
20010315"/>
    <dsig:SignatureMethod Algorithm="ALGORITHM"/>
    <dsig:Reference Id="signed-data-reference-SIGDATAREF" URI="#signed-data-object-
SIGDATAOBJURI">
      <dsig:Transforms>
        <dsig:Transform Algorithm="http://www.w3.org/2002/06/xmldsig-filter2">
          <xpf:XPath xmlns:xpf="http://www.w3.org/2002/06/xmldsig-filter2"
Filter="intersect">id('signed-data-object-SIGDATAOBJURI')/node()</xpf:XPath>
        </dsig:Transform>
        <dsig:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#base64"/>
      </dsig:Transforms>
      <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
      <dsig:DigestValue>DIGESTVALUESIGNEDDATA</dsig:DigestValue>
    </dsig:Reference>
    <dsig:Reference Id="etsi-data-reference-ETSIDATAREF"
Type="http://uri.etsi.org/01903/v1.1.1#SignedProperties" URI="#etsi-data-object-
ETSIDATAOBJURI">
      <dsig:Transforms>
        <dsig:Transform Algorithm="http://www.w3.org/2002/06/xmldsig-filter2">
          <xpf:XPath xmlns:xpf="http://www.w3.org/2002/06/xmldsig-filter2"
Filter="intersect">id('etsi-data-object-ETSIDATAOBJURI')/node()</xpf:XPath>
        </dsig:Transform>
      </dsig:Transforms>
      <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
      <dsig:DigestValue>DIGESTVALUESIGNEDPROPERTIES</dsig:DigestValue>
    </dsig:Reference>
  </dsig:SignedInfo>
  <dsig:SignatureValue>SIGNATUREVALUE</dsig:SignatureValue>
  <dsig:KeyInfo>
    <dsig:X509Data>
      <dsig:X509Certificate>X509CERTIFICATE</dsig:X509Certificate>
    </dsig:X509Data>
  </dsig:KeyInfo>
  <dsig:Object Id="signed-data-object-SIGDATAOBJURI">
    <sl:Base64Content>BASE64CONTENT</sl:Base64Content>
  </dsig:Object>
```

```
 <dsig:Object Id="etsi-data-object-ETSIDATAOBJURI">
  <etsi:QualifyingProperties xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
xmlns:etsi="http://uri.etsi.org/01903/v1.1.1#" Target="#signature-SIGID">
   <etsi:SignedProperties>
    <etsi:SignedSignatureProperties>
     <etsi:SigningTime>SIGNINGTIME</etsi:SigningTime>
     <etsi:SigningCertificate>
      <etsi:Cert>
       <etsi:CertDigest>
        <etsi:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
        <etsi:DigestValue>DIGESTVALUEX509CERTIFICATE</etsi:DigestValue>
       </etsi:CertDigest>
       <etsi:IssuerSerial>
        <dsig:X509IssuerName>X509ISSUERNAME</dsig:X509IssuerName>
        <dsig:X509SerialNumber>X509SERIALNUMBER</dsig:X509SerialNumber>
       </etsi:IssuerSerial>
      </etsi:Cert>
     </etsi:SigningCertificate>
     <etsi:SignaturePolicyIdentifier>
      <etsi:SignaturePolicyImplied/>
     </etsi:SignaturePolicyIdentifier>
    </etsi:SignedSignatureProperties>
    <etsi:SignedDataObjectProperties>
     <etsi:DataObjectFormat ObjectReference="#signed-data-reference-SIGDATAREF">
      <etsi:MimeType>text/plain</etsi:MimeType>
     </etsi:DataObjectFormat>
    </etsi:SignedDataObjectProperties>
   </etsi:SignedProperties>
  </etsi:QualifyingProperties>
 </dsig:Object>
</dsig:Signature>
```

Die vom Layout vorgesehenen Variabilitäten (Variablen) werden in Großbuchstaben und umrandet ausgewiesen. Die nachfolgenden Abschnitte definieren diese im Detail.

Dieses Profil legt durch das damit festgelegte XML-Signaturlayout die folgenden wesentlichen XML-Signatur-Eigenschaften implizit fest:

1. Kanonisierungsmethode: http://www.w3.org/TR/2001/REC-xml-c14n-20010315
2. Digest-Methode: http://www.w3.org/2000/09/xmldsig#sha1
3. Qualifying Properties: nach ETSI (http://uri.etsi.org/01903/v1.1.1#)

5.2.3.1 SIGID

Anstelle der Variable **SIGID** MUSS im XML-Signaturlayout der aus den Signaturparametern zusammengesetzte Wert `<ParamSigID>` eingesetzt werden.

5.2.3.2 ALGORITHM

Legt den Signaturalgorithmus fest. Es MUSS einer der folgenden Signaturalgorithmen verwendet werden, in Abhängigkeit der Eigenschaften des Signatur-Schlüssels:

- für ECDSA-Schlüssel: `http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha1`
- für RSA-Schlüssel: `http://www.w3.org/2000/09/xmldsig#rsa-sha1`

Der zum Signaturschlüssel passende Identifikator (URI) MUSS im XML-Signaturlayout anstelle der Variable **ALGORITHM** eingesetzt werden.

5.2.3.3 SIGDATAREF

Anstelle der Variable **SIGDATAREF** MUSS im XML-Signaturlayout der aus den Signaturparametern zusammengesetzte Wert `<ParamSigDataRef>` eingesetzt werden.

5.2.3.4 SIGDATAOBJURI

Anstelle der Variable **SIGDATAOBJURI** MUSS im XML-Signaturlayout der aus den Signaturparametern zusammengesetzte Wert `<ParamSigDataObjURI>` eingesetzt werden.

5.2.3.5 DIGESTVALUESIGNEDDATA

Anstelle der Variable **DIGESTVALUESIGNEDDATA** MUSS im XML-Signaturlayout der Hash-Wert der ersten XML-Signaturreferenz eingesetzt werden.

5.2.3.6 ETSIDATAREF

Anstelle der Variable **ETSIDATAREF** MUSS im XML-Signaturlayout der aus den Signaturparametern zusammengesetzte Wert `<ParamEtsiDataRef>` eingesetzt werden.

5.2.3.7 ETSIDATAOBJURI

Anstelle der Variable **ETSIDATAOBJURI** MUSS im XML-Signaturlayout der aus den Signaturparametern zusammengesetzte Wert `<ParamEtsiDataObjURI>` eingesetzt werden.

5.2.3.8 DIGESTVALUESIGNEDPROPERTIES

Anstelle der Variable **DIGESTVALUESIGNEDPROPERTIES** MUSS im XML-Signaturlayout der Hash-Wert der zweiten XML-Signaturreferenz eingesetzt werden.

5.2.3.9 SIGNATUREVALUE

Anstelle der Variable **SIGNATUREVALUE** MUSS im XML-Signaturlayout der Signaturwert eingesetzt werden.

5.2.3.10 X509CERTIFICATE

Anstelle der Variable **X509CERTIFICATE** MUSS im XML-Signaturlayout das Base64-kodierte Signaturzertifikat (X509-Zertifikat, kodiert gem. Abschnitt 4.4.4 in [4]) eingesetzt werden.

5.2.3.11 DSIGOBJECT

Wird eine Signatur-Methode angewandt, die eine enveloping XML-Signatur erwirkt, so MUSS das folgende Fragment anstelle der Variable **DSIGOBJECT** im XML-Signaturlayout eingesetzt werden:

```
<dsig:Object Id="signed-data-object-SIGDATAOBJURI">
   <dsig:Base64Content>BASE64CONTENT</dsig:Base64Content>
</dsig:Object>
```

In diesem Fragment MUSS anstelle der Variable **SIGDATAOBJURI** der aus den Signaturparametern zusammengesetzte Wert `<ParamSigDataObjURI>` eingesetzt werden.

In diesem Fragment MUSS anstelle der Variable **BASE64CONTENT** der Base64-kodierte, zu signierende Datenstrom (gem. angewandter Signaturmethode) eingesetzt werden.

Wird eine Signatur-Methode angewandt, die eine detached XML-Signatur erwirkt, so ist kein XML-Signaturobjekt für die signierten Daten notwendig. Die Variable **DSIGOBJECT** im XML-Signaturlayout MUSS ersatzlos entfernt werden.

5.2.3.12 SIGNINGTIME

Anstelle der Variable **SIGNINGTIME** MUSS der Signaturzeitpunkt gem. [5] eingesetzt werden.

5.2.3.13 DIGESTVALUEX509CERTIFICATE

Anstelle der Variable **DIGESTVALUEX509CERTIFICATE** MUSS der Hash-Wert des Signaturzertifikates (lt. [5]) eingesetzt werden.

5.2.3.14 X509ISSUERNAME

Anstelle der Variable **X509ISSUERNAME** MUSS der Name des Ausstellers des Signaturzertifikates (lt. [5]) eingesetzt werden.

5.2.3.15 X509SERIALNUMBER

Anstelle der Variable **X509SERIALNUMBER** MUSS die Seriennummer des Signaturzertifikates (lt. [5]) eingesetzt werden.

5.3 Signaturparameter-Profil `etsi-bka-1.0`

5.3.1 Charakteristik

Parameter-Kennzeichnung:	`etsi-bka-1.0`
Signaturerstellungskomponente:	dieses Profil MUSS für die Generallizenz der Bürgerkartenumgebung (IT-Solution trustDesk-Basic), ab Version 2.7.5 angewendet werden.
Einschränkungen bzgl. Signatur-methoden:	keine, kann mit allen Signaturmethoden verwendet werden
Verwendbarkeit:	EMPFOHLEN

5.3.2 Signaturparameter

Für dieses Signaturparameter-Profil wurden Signaturparameter definiert. Diese MÜSSEN nach der vorangestellten Kennzeichnung des Profils in die Signatur-Repräsentation eingefügt werden.

Für die Formulierung der Signaturparameter MUSS die Definition gem. Abschnitt 2.2 angewandt werden. Es gilt bei diesem Signaturparameter-Profil jedoch die Maßgabe, dass das Feld `<SIGDEV_PROF>` den Wert `etsi-bka-1.0` haben MUSS.

Die Signaturparameter dieses Profils repräsentieren 5 Einzelwerte. Dabei MUSS als Parameter Teil 1 (Feld `PARAM_L1` der Signaturparameter, siehe 2.2) als konstanter Präfix für alle 5 Einzelwerte herangezogen werden. Der Parameter Teil 2 (Feld `PARAM_L2` der Signaturparameter, siehe 2.2) enthält selbst 5 Einzelwerte, die jeweils mit einem Bindestrich separiert werden müssen. Es gilt für Parameter Teil 2 folgende Ausformung (in Ergänzung zur Definition aus Abschnitt 2.2):

```
<PARAM_L2> :: = <WERT_1>"-"<WERT_2>"-"<WERT_3>"-"<WERT_4>"-"<WERT_5>
   <WERT_1> :: = <CHAR>
   <WERT_2> :: = <CHAR>
   <WERT_3> :: = <CHAR>
   <WERT_4> :: = <CHAR>
   <WERT_5> :: = <CHAR>
```

Daraus MÜSSEN folgende Einzelwerte gebildet werden:

```
<ParamSigID>           ::= <PARAM_L1>"-"<WERT_1>
<ParamSigDataRef>      ::= "0-"<PARAM_L1>"-"<WERT_2>
<ParamSigDataObjURI>   ::= "0-"<PARAM_L1>"-"<WERT_3>
<ParamEtsiDataRef>     ::= "0-"<PARAM_L1>"-"<WERT_4>
<ParamEtsiDataObjURI>  ::= "0-"<PARAM_L1>"-"<WERT_5>
```

Dieser Werte werden mehrfach im Signaturlayout referenziert und verwendet. Wird eine Signatur-Methode angewandt, die eine detached XML-Signatur erzeugt, so wird der Wert `<ParamSigDataObjURI>` nicht benötigt. Die Signaturparameter lassen die Bildung dieses Wertes jedoch dennoch zu.

Die Verwendung dieser Einzelwerte wird in Abschnitt 5.2.3 festgelegt.

Nachfolgend ein Beispiel zur Bildung der Einzelwerte auf Basis übergebener Signaturparameter (spezifisch für das vorliegende Signaturparameter-Profil):

Signaturparameter lt. Signaturrepräsentation:

```
etsi-bka-1.0@1200412799-27800484@23524-22018-0-24095-30271
```

Daraus ergeben sich:

```
<ParamSigID>          = 1200412799-27800484-23524
<ParamSigDataRef>     = 0-1200412799-27800484-22018
<ParamSigDataObjURI>  = 0-1200412799-27800484-0
<ParamEtsiDataRef>    = 0-1200412799-27800484-24095
<ParamEtsiDataObjURI> = 0-1200412799-27800484-30271
```

5.3.3 Signaturlayout

Die Signaturen einer Bürgerkartenumgebung enthalten zeitabhängige Varianzen. Diese sind vor allem eine Reihe von XML-Attributen (ID-Attribute) die zur Referenzierung von XML-Elementen/-Knoten herangezogen werden. Diese variablen Attribute MÜSSEN in Form der Signaturparameter innerhalb der Signatur-Repräsentation verwaltet und bei der Rekonstruktion der Signatur entsprechend berücksichtigt werden.

Eine XML-Signatur, die diesem Parameter-Profil entspricht, MUSS dem folgenden Layout entsprechen:

```
<dsig:Signature xmlns:dsig="http://www.w3.org/2000/09/xmldsig#" Id="signature-SIGID">
  <dsig:SignedInfo>
    <dsig:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-
20010315"/>
    <dsig:SignatureMethod Algorithm="ALGORITHM"/>
    <dsig:Reference Id="signed-data-reference-SIGDATAREF" URI="REFERENCE">
      TRANSFORMS
      <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
      <dsig:DigestValue>DIGESTVALUESIGNEDDATA</dsig:DigestValue>
    </dsig:Reference>
    <dsig:Reference Id="etsi-data-reference-ETSIDATAREF"
Type="http://uri.etsi.org/01903/v1.1.1#SignedProperties"
URI="#xmlns(etsi=http://uri.etsi.org/01903/v1.1.1%23)%20xpointer(id('etsi-data-object-
ETSIDATAOBJURI')/child::etsi:QualifyingProperties/child::etsi:SignedProperties)">
      <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
      <dsig:DigestValue>DIGESTVALUESIGNEDPROPERTIES</dsig:DigestValue>
    </dsig:Reference>
  </dsig:SignedInfo>
  <dsig:SignatureValue>SIGNATUREVALUE</dsig:SignatureValue>
  <dsig:KeyInfo>
    <dsig:X509Data>
      <dsig:X509Certificate>X509CERTIFICATE</dsig:X509Certificate>
    </dsig:X509Data>
  </dsig:KeyInfo>
  DSIGOBJECT
  <dsig:Object Id="etsi-data-object-ETSIDATAOBJURI">
    <etsi:QualifyingProperties xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
xmlns:etsi="http://uri.etsi.org/01903/v1.1.1#" Target="#signature-SIGID">
      <etsi:SignedProperties xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
xmlns:etsi="http://uri.etsi.org/01903/v1.1.1#">
        <etsi:SignedSignatureProperties>
          <etsi:SigningTime>SIGNINGTIME</etsi:SigningTime>
```

```
      <etsi:SigningCertificate>
       <etsi:Cert>
        <etsi:CertDigest>
         <etsi:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
         <etsi:DigestValue>DIGESTVALUEX509CERTIFICATE</etsi:DigestValue>
        </etsi:CertDigest>
        <etsi:IssuerSerial>
         <dsig:X509IssuerName>X509ISSUERNAME</dsig:X509IssuerName>
         <dsig:X509SerialNumber>X509SERIALNUMBER</dsig:X509SerialNumber>
        </etsi:IssuerSerial>
       </etsi:Cert>
      </etsi:SigningCertificate>
      <etsi:SignaturePolicyIdentifier>
       <etsi:SignaturePolicyImplied/>
      </etsi:SignaturePolicyIdentifier>
     </etsi:SignedSignatureProperties>
     <etsi:SignedDataObjectProperties>
      <etsi:DataObjectFormat ObjectReference="#signed-data-reference-SIGDATAREF">
       <etsi:MimeType>MIMETYPE</etsi:MimeType>
      </etsi:DataObjectFormat>
     </etsi:SignedDataObjectProperties>
    </etsi:SignedProperties>
   </etsi:QualifyingProperties>
  </dsig:Object>
</dsig:Signature>
```

Die vom Layout vorgesehenen Variabilitäten (Variablen) werden in Großbuchstaben und umrandet ausgewiesen. Die nachfolgenden Abschnitte definieren diese im Detail.

Dieses Profil legt durch das damit festgelegte XML-Signaturlayout die folgenden wesentlichen XML-Signatur-Eigenschaften implizit fest:

1. Kanonisierungsmethode: `http://www.w3.org/TR/2001/REC-xml-c14n-20010315`
2. Digest-Methode: `http://www.w3.org/2000/09/xmldsig#sha1`
3. Qualifying Properties: nach ETSI (`http://uri.etsi.org/01903/v1.1.1#`)

5.3.3.1 SIGID

Anstelle der Variable **SIGID** MUSS im XML-Signaturlayout der aus den Signaturparametern zusammengesetzte Wert `<ParamSigID>` eingesetzt werden.

5.3.3.2 ALGORITHM

Legt den Signaturalgorithmus fest. Es MUSS einer der folgenden Signaturalgorithmen verwendet werden, in Abhängigkeit der Eigenschaften des Signatur-Schlüssels:

- für ECDSA-Schlüssel: `http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha1`
- für RSA-Schlüssel: `http://www.w3.org/2000/09/xmldsig#rsa-sha1`

Der zum Signaturschlüssel passende Identifikator (URI) MUSS im XML-Signaturlayout anstelle der Variable **ALGORITHM** eingesetzt werden.

5.3.3.3 SIGDATAREF

Anstelle der Variable **SIGDATAREF** MUSS im XML-Signaturlayout der aus den Signaturparametern zusammengesetzte Wert `<ParamSigDataRef>` eingesetzt werden.

5.3.3.4 SIGDATAOBJURI

Anstelle der Variable **SIGDATAOBJURI** MUSS im XML-Signaturlayout der aus den Signaturparametern zusammengesetzte Wert `<ParamSigDataObjURI>` eingesetzt werden.

Dieser Wert wird mehrfach im Signaturlayout referenziert und verwendet. Wird eine Signatur-Methode angewandt, die eine detached XML-Signatur erzeugt, so wird dieser Wert nicht benötigt. Die Signaturparameter lassen die Bildung dieses Wertes jedoch dennoch zu.

5.3.3.5 REFERENCE

Wird eine Signatur-Methode angewandt, die eine enveloping XML-Signatur erwirkt, so MUSS das folgende Fragment anstelle der Variable **REFERENCE** im XML-Signaturlayout eingesetzt werden:

```
#signed-data-object-SIGDATAOBJURI
```

In diesem Fragment MUSS anstelle der Variable **SIGDATAOBJURI** der aus den Signaturparametern zusammengesetzte Wert `<ParamSigDataObjURI>` eingesetzt werden.

Beispiel: `#signed-data-object-0-1155648477-25748375-22389`

Wird eine Signatur-Methode angewandt, die eine detached XML-Signatur erzeugt, so MUSS das folgende Fragment anstelle der Variable **REFERENCE** im XML-Signaturlayout eingesetzt werden:

```
urn:Document
```

5.3.3.6 TRANSFORMS

Wird eine Signatur-Methode angewandt, die eine enveloping XML-Signatur erzeugt, so MUSS das folgende Fragment anstelle der Variable **TRANSFORMS** im XML-Signaturlayout eingesetzt werden:

```
<dsig:Transforms>
   <dsig:Transform Algorithm="http://www.w3.org/2002/06/xmldsig-filter2">
      <xpf:XPath xmlns:xpf="http://www.w3.org/2002/06/xmldsig-filter2"
Filter="intersect">id('signed-data-object-SIGDATAOBJURI')/node()</xpf:XPath>
   </dsig:Transform>
   <dsig:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#base64"/>
</dsig:Transforms>
```

In diesem Fragment MUSS anstelle der Variable **SIGDATAOBJURI** der aus den Signaturparametern zusammengesetzte Wert `<ParamSigDataObjURI>` eingesetzt werden.

Wird eine Signatur-Methode angewandt, die eine detached XML-Signatur erwirkt, so ist kein Transformationspfad einzufügen. In diesem Fall DARF ein Transformationspfad NICHT im XML-Signaturlayout eingesetzt werden. Die Variable **TRANSFORMS** im XML-Signaturlayout MUSS ersatzlos entfernt werden.

5.3.3.7 DIGESTVALUESIGNEDDATA

Anstelle der Variable **DIGESTVALUESIGNEDDATA** MUSS im XML-Signaturlayout der Hash-Wert der ersten XML-Signaturreferenz eingesetzt werden.

5.3.3.8 ETSIDATAREF

Anstelle der Variable **ETSIDATAREF** MUSS im XML-Signaturlayout der aus den Signaturparametern zusammengesetzte Wert `<ParamEtsiDataRef>` eingesetzt werden.

5.3.3.9 ETSIDATAOBJURI

Anstelle der Variable **ETSIDATAOBJURI** MUSS im XML-Signaturlayout der aus den Signaturparametern zusammengesetzte Wert `<ParamEtsiDataObjURI>` eingesetzt werden.

5.3.3.10 DIGESTVALUESIGNEDPROPERTIES

Anstelle der Variable **DIGESTVALUESIGNEDPROPERTIES** MUSS im XML-Signaturlayout der Hash-Wert der zweiten XML-Signaturreferenz eingesetzt werden.

5.3.3.11 SIGNATUREVALUE

Anstelle der Variable **SIGNATUREVALUE** MUSS im XML-Signaturlayout der Signaturwert eingesetzt werden.

5.3.3.12 X509CERTIFICATE

Anstelle der Variable **X509CERTIFICATE** MUSS im XML-Signaturlayout das Base64-kodierte Signaturzertifikat (X509-Zertifikat, kodiert gem. Abschnitt 4.4.4 in [4]) eingesetzt werden.

5.3.3.13 DSIGOBJECT

Wird eine Signatur-Methode angewandt, die eine enveloping XML-Signatur erwirkt, so MUSS das folgende Fragment anstelle der Variable **DSIGOBJECT** im XML-Signaturlayout eingesetzt werden:

```
<dsig:Object Id="signed-data-object-SIGDATAOBJURI">
   <dsig:Base64Content>BASE64CONTENT</dsig:Base64Content>
</dsig:Object>
```

In diesem Fragment MUSS anstelle der Variable **SIGDATAOBJURI** der aus den Signaturparametern zusammengesetzte Wert `<ParamSigDataObjURI>` eingesetzt werden.

In diesem Fragment MUSS anstelle der Variable **BASE64CONTENT** der Base64-kodierte, zu signierende Datenstrom (gem. angewandter Signaturmethode) eingesetzt werden.

Wird eine Signatur-Methode angewandt, die eine detached XML-Signatur erwirkt, so ist kein XML-Signaturobjekt für die signierten Daten notwendig. Die Variable **DSIGOBJECT** im XML-Signaturlayout MUSS ersatzlos entfernt werden.

5.3.3.14 SIGNINGTIME

Anstelle der Variable **SIGNINGTIME** MUSS der Signaturzeitpunkt gem. [5] eingesetzt werden.

5.3.3.15 DIGESTVALUEX509CERTIFICATE

Anstelle der Variable **DIGESTVALUEX509CERTIFICATE** MUSS der Hash-Wert des Signaturzertifikates (lt. [5]) eingesetzt werden.

5.3.3.16 X509ISSUERNAME

Anstelle der Variable **X509ISSUERNAME** MUSS der Name des Ausstellers des Signaturzertifikates (lt. [5]) eingesetzt werden.

5.3.3.17 X509SERIALNUMBER

Anstelle der Variable **X509SERIALNUMBER** MUSS die Seriennummer des Signaturzertifikates (lt. [5]) eingesetzt werden.

5.3.3.18 MIMETYPE

Anstelle der Variable **MIMETYPE** MUSS der MIME-Type der zu signierenden Daten eingesetzt werden. Der MIME-Type ist von der angewandten Signaturmethode abhängig und MUSS im Zuge der Definition der Signaturmethode festgelegt werden.

5.4 Signaturparameter-Profil `etsi-moc-1.0`

5.4.1 Charakteristik

Parameter-Kennzeichnung: `etsi-moc-1.0`

Signaturerstellungskomponente: dieses Profil MUSS für die Open Source BKU "MOCCA" verwendet werden.

Einschränkungen bzgl. Signaturmethoden: `urn:pdfsigfilter:bka.gv.at:text:v1.1.0`, `urn:pdfsigfilter:bka.gv.at:binaer:v1.1.0`

Verwendbarkeit: EMPFOHLEN

5.4.2 Signaturparameter

Für dieses Signaturparameter-Profil wurden Signaturparameter definiert. Diese MÜSSEN nach der vorangestellten Kennzeichnung des Profils in die Signatur-Repräsentation eingefügt werden.

Für die Formulierung der Signaturparameter MUSS die Definition gem. Abschnitt 2.2 angewandt werden. Es gilt bei diesem Signaturparameter-Profil jedoch die Maßgabe, dass das Feld `<SIGDEV_PROF>` den Wert `etsi-moc-1.0` haben MUSS.

Der Signaturparameter `PARAM_L1` (siehe Abschnitt 2.2) stellt einen jeweils innerhalb einer Signatur konstanten Wert dar, der für die Bildung der 7 unten beschriebenen Einzelwerte herangezogen werden MUSS. Der Parameter `PARAM_L2` wird nicht verwendet und entfällt zusammen mit dem Delimiter-Zeichen "`@`".

Aus dem Signaturparameter `PARAM_L1` MÜSSEN folgende Einzelwerte gebildet werden:

```
<ParamSigId>                  ::= "Signature-"<PARAM_L1>"-1"
<ParamSignedInfoId>           ::= "SignedInfo-"<PARAM_L1>"-1"
<ParamSigDataRefId>           ::= "Reference-"<PARAM_L1>"-1"
<ParamEtsiDataRefId>          ::= "Reference-"<PARAM_L1>"-2"
<ParamSigValueId>             ::= "SignatureValue-"<PARAM_L1>"-1"
<ParamEtsiSignedPropertiesId> ::= "SignedProperties-"<PARAM_L1>"-1"
<ParamEtsiDataObjId>          ::= "Object-"<PARAM_L1>"-1"
```

Diese Werte werden mehrfach im Signaturlayout referenziert und verwendet.

Die Verwendung dieser Einzelwerte wird in Abschnitt 5.4.3 festgelegt.

Nachfolgend ein Beispiel zur Bildung der Einzelwerte auf Basis übergebener Signaturparameter (spezifisch für das vorliegende Signaturparameter-Profil):

Signaturparameter lt. Signaturrepräsentation:

```
etsi-moc-1.0@b2e01c95
```

Daraus ergeben sich:

```
<ParamSigId>                  = Signature-b2e01c95-1
<ParamSignedInfoId>           = SignedInfo-b2e01c95-1
<ParamSigDataRefId>           = Reference-b2e01c95-1
<ParamEtsiDataRefId>          = Reference-b2e01c95-2
<ParamSigValueId>             = SignatureValue-b2e01c95-1
<ParamEtsiSignedPropertiesId> = SignedProperties-b2e01c95-1
<ParamEtsiDataObjId>          = Object-b2e01c95-1
```

5.4.3 Signaturlayout

Die Signaturen einer Bürgerkartenumgebung enthalten zeitabhängige Varianzen. Diese sind vor allem eine Reihe von XML-Attributen (ID-Attribute) die zur Referenzierung von XML-Elementen/-Knoten herangezogen werden. Diese variablen Attribute MÜSSEN in Form der Signaturparameter innerhalb der Signatur-Repräsentation verwaltet und bei der Rekonstruktion der Signatur entsprechend berücksichtigt werden.

Eine XML-Signatur, die diesem Parameter-Profil entspricht, MUSS dem folgenden Layout entsprechen:

```
<dsig:Signature xmlns:dsig="http://www.w3.org/2000/09/xmldsig#" Id="SIGID">
  <dsig:SignedInfo Id="SIGNEDINFOID">
    <dsig:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
    <dsig:SignatureMethod Algorithm="ALGORITHM"/>
    <dsig:Reference Id="SIGDATAREFID" URI="urn:Document">
      <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
      <dsig:DigestValue>DIGESTVALUESIGNEDDATA</dsig:DigestValue>
    </dsig:Reference>
    <dsig:Reference Id="ETSIDATAREFID"
Type="http://uri.etsi.org/01903/v1.1.1#SignedProperties"
URI="#xmlns(xades=http://uri.etsi.org/01903/v1.1.1%23)%20xpointer(id('ETSIDATAOBJID')/
child::xades:QualifyingProperties/child::xades:SignedProperties)">
      <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
      <dsig:DigestValue>DIGESTVALUESIGNEDPROPERTIES</dsig:DigestValue>
    </dsig:Reference>
  </dsig:SignedInfo>
  <dsig:SignatureValue Id="SIGVALUEID">SIGNATUREVALUE</dsig:SignatureValue>
  <dsig:KeyInfo>
    <dsig:X509Data>
      <dsig:X509Certificate>X509CERTIFICATE</dsig:X509Certificate>
    </dsig:X509Data>
  </dsig:KeyInfo>
  <dsig:Object Id="ETSIDATAOBJID">
    <QualifyingProperties xmlns="http://uri.etsi.org/01903/v1.1.1#"
xmlns:ns2="http://www.w3.org/2000/09/xmldsig#">
      <SignedProperties xmlns="http://uri.etsi.org/01903/v1.1.1#"
xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
xmlns:ns2="http://www.w3.org/2000/09/xmldsig#" Id="ETSISIGNEDPROPERTIESID">
        <SignedSignatureProperties>
          <SigningTime>SIGNINGTIME</SigningTime>
          <SigningCertificate>
            <Cert>
              <CertDigest>
                <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
                <DigestValue>DIGESTVALUEX509CERTIFICATE</DigestValue>
              </CertDigest>
              <IssuerSerial>
                <ns2:X509IssuerName>X509ISSUERNAME</ns2:X509IssuerName>
                <ns2:X509SerialNumber>X509SERIALNUMBER</ns2:X509SerialNumber>
              </IssuerSerial>
            </Cert>
          </SigningCertificate>
          <SignaturePolicyIdentifier>
            <SignaturePolicyImplied/>
          </SignaturePolicyIdentifier>
        </SignedSignatureProperties>
        <SignedDataObjectProperties>
          <DataObjectFormat ObjectReference="#SIGDATAREFID">
            <MimeType>MIMETYPE</MimeType>
          </DataObjectFormat>
        </SignedDataObjectProperties>
      </SignedProperties>
    </QualifyingProperties>
  </dsig:Object>
</dsig:Signature>
```

Die vom Layout vorgesehenen Variabilitäten (Variablen) werden in Großbuchstaben und umrandet ausgewiesen. Die nachfolgenden Abschnitte definieren diese im Detail.

Dieses Profil legt durch das damit festgelegte XML-Signaturlayout die folgenden wesentlichen XML-Signatur-Eigenschaften implizit fest:

1. Kanonisierungsmethode: `http://www.w3.org/2001/10/xml-exc-c14n#`
2. Digest-Methode: `http://www.w3.org/2000/09/xmldsig#sha1`
3. Qualifying Properties: nach ETSI (`http://uri.etsi.org/01903/v1.1.1#`)

5.4.3.1 SIGID

Anstelle der Variable **SIGID** MUSS im XML-Signaturlayout der aus dem Signaturparameter zusammengesetzte Wert `<ParamSigId>` eingesetzt werden.

5.4.3.2 SIGNEDINFOID

Anstelle der Variable **SIGNEDINFOID** MUSS im XML-Signaturlayout der aus dem Signaturparameter zusammengesetzte Wert `<ParamSignedInfoId>` eingesetzt werden.

5.4.3.3 ALGORITHM

Legt den Signaturalgorithmus fest. Es MUSS einer der folgenden Signaturalgorithmen verwendet werden, in Abhängigkeit der Eigenschaften des Signatur-Schlüssels:

- für ECDSA-Schlüssel: `http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha1`
- für RSA-Schlüssel: `http://www.w3.org/2000/09/xmldsig#rsa-sha1`

Der zum Signaturschlüssel passende Identifikator (URI) MUSS im XML-Signaturlayout anstelle der Variable **ALGORITHM** eingesetzt werden.

5.4.3.4 SIGDATAREFID

Anstelle der Variable **SIGDATAREFID** MUSS im XML-Signaturlayout der aus den Signaturparametern zusammengesetzte Wert `<ParamSigDataRefId>` eingesetzt werden.

5.4.3.5 DIGESTVALUESIGNEDDATA

Anstelle der Variable **DIGESTVALUESIGNEDDATA** MUSS im XML-Signaturlayout der Base64-kodierte Hash-Wert der ersten XML-Signaturreferenz eingesetzt werden.

5.4.3.6 ETSIDATAREFID

Anstelle der Variable **ETSIDATAREFID** MUSS im XML-Signaturlayout der aus den Signaturparametern zusammengesetzte Wert `<ParamEtsiDataRefId>` eingesetzt werden.

5.4.3.7 ETSIDATAOBJID

Anstelle der Variable **ETSIDATAOBJID** MUSS im XML-Signaturlayout der aus den Signaturparametern zusammengesetzte Wert `<ParamEtsiDataObjId>` eingesetzt werden.

5.4.3.8 DIGESTVALUESIGNEDPROPERTIES

Anstelle der Variable **DIGESTVALUESIGNEDPROPERTIES** MUSS im XML-Signaturlayout der Base64-kodierte Hash-Wert der zweiten XML-Signaturreferenz eingesetzt werden.

5.4.3.9 SIGVALUEID

Anstelle der Variable **SIGVALUEID** MUSS im XML-Signaturlayout der aus den Signaturparametern zusammengesetzte Wert `<ParamSigValueId>` eingesetzt werden.

5.4.3.10 SIGNATUREVALUE

Anstelle der Variable **SIGNATUREVALUE** MUSS im XML-Signaturlayout der Base64-kodierte Signaturwert eingesetzt werden.

5.4.3.11 X509CERTIFICATE

Anstelle der Variable **X509CERTIFICATE** MUSS im XML-Signaturlayout das Base64-kodierte Signaturzertifikat (X509-Zertifikat, kodiert gem. Abschnitt 4.4.4 in [4]) eingesetzt werden.

5.4.3.12 ETSISIGNEDPROPERTIESID

Anstelle der Variable **ETSISIGNEDPROPERTIESID** MUSS im XML-Signaturlayout der aus den Signaturparametern zusammengesetzte Wert `<ParamEtsiSignedPropertiesId>` eingesetzt werden.

5.4.3.13 SIGNINGTIME

Anstelle der Variable **SIGNINGTIME** MUSS der Signaturzeitpunkt gem. [5] eingesetzt werden.

5.4.3.14 DIGESTVALUEX509CERTIFICATE

Anstelle der Variable **DIGESTVALUEX509CERTIFICATE** MUSS der Base64-kodierte Hash-Wert des Signaturzertifikates (lt. [5]) eingesetzt werden.

5.4.3.15 X509ISSUERNAME

Anstelle der Variable **X509ISSUERNAME** MUSS der Name des Ausstellers des Signaturzertifikates (lt. [5]) eingesetzt werden.

5.4.3.16 X509SERIALNUMBER

Anstelle der Variable **X509SERIALNUMBER** MUSS die Seriennummer des Signaturzertifikates (lt. [5]) eingesetzt werden.

5.4.3.17 MIMETYPE

Anstelle der Variable **MIMETYPE** MUSS der MIME-Type der zu signierenden Daten eingesetzt werden. Der MIME-Type ist von der angewandten Signaturmethode abhängig und MUSS im Zuge der Definition der Signaturmethode festgelegt werden.

Referenzen

[1] ISO-19005-1:2005: Document management - Electronic document file format for long-term preservation - Part 1: Use of PDF 1.4 (PDF/A-1). ISO Standard 19005-1:2005.

[2] Adobe Corporation: PDF Reference, third edition - Adobe portable document format version 1.4, ISBN 0-201-75839-3, December 2001, Addison-Wesley.

[3] Adobe Corporation: PDF Reference, fifth edition - Adobe Portable Document Format version 1.6.

[4] Eastlake, Donald, Reagle, Joseph und Solo, David: XML-Signature Syntax and Processing. W3C Recommendation, Februar 2002. Abgerufen aus dem World Wide Web am 14.05.2004 unter http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/.

[5] ETSI XML Advanced Electronic Signatures (XAdES): ETSI TS 101 903 V1.2.2 (2004-04), Technical Specification.

[6] Hollosi, Karlinger, et.al.: Die österreichische Bürgerkarte , Spezifikationsframework Version 1.2, März 2005. Abgerufen aus dem World Wide Web am 18.01.2008 unter http://www.buergerkarte.at/konzept/securitylayer/spezifikation/aktuell/.

[7] Stabsstelle IKT-Strategie des Bundes: Spezifikation Module für Online Anwendungen – SP und SS, Version 1.3.0, vom 28. August 2005. Abgerufen aus dem World Wide Web am 10.01.2008 unter http://egovlabs.gv.at/docman/view.php/6/20/MOA-SPSS-1.3.pdf.

[8] Rössler: Layout Amtssignatur – Spezifikation. Version 1.0.0 vom 25.06.2007.

[9] RFC 2234: Augmented BNF for Syntax Specifications: ABNF