

PDF Signatur/Amtssignatur Spezifikation

Version 2.1.1

Inhalt

PDF Signatur/Amtssignatur	1
Spezifikation	1
Dokument-Historie	3
Schlüsselwörter	3
Begriffserklärungen	4
1 Einleitung	6
2 Charakteristika von PDF-Amtssignaturen	7
2.1 Methode	7
2.2 Parameter	8
2.3 Anforderungen an PDF Dokumente	9
3 Repräsentation einer PDF-Amtssignatur	9
4 Definierte Signaturmethoden	11
4.1 Textuelle Signatur, Version 1.0.0	11
4.2 Textuelle Signatur, Version 1.1.0	15
4.3 Binäre Signatur, Version 1.0.0	19
4.4 Binäre Signatur, Version 1.1.0	29
5 Definierte Signaturparameter	38
5.1 Default Signaturparameter-Profil	38
5.2 Default Signaturparameter-Profil für BKU	42
5.3 Signaturparameter-Profil etsi-bka-1.0	46
5.4 Signaturparameter-Profil etsi-moc-1.0	51
Referenzen	55

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 <i>sichtbare</i> 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.

1 **Begriffserklärungen**

2 ***Binäre Signatur:***

3 Eine binäre Signatur signiert das gesamte Dokument in binärer Repräsentation.

4 ***Detached Signatures:***

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

8 *Signature, Detached*

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

14 ***Enveloping Signatures:***

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

16 *Signature, Enveloping*

17 *The signature is over content found within an Object element of the signature*
18 *itself. The Object (or its content) is identified via a Reference (via a URI fragment*
19 *identifier or transform).*

20 ***Mehrfachsignatur:***

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

29 ***Portable Document Format (PDF):***

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

36 ***Prüfvorgang:***

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

39 ***Signaturblock:***

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

42

43 **Signaturvorgang:**

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

46 **Textuelle Signatur:**

47 Eine textuelle Signatur signiert den textuellen Inhalt eines Dokuments.

48 **Signaturattribute:**

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

52

53 **1 Einleitung**

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

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

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

- 67 - textuelle PDF-Signatur
- 68 - binäre PDF-Signatur

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

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

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

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

86

87 2 Charakteristika von PDF-Amtssignaturen

88 PDF-Amtssignaturen werden durch zwei Identifikationsbegriffe charakterisiert:

- 89 - Signaturmethode (Methode)
- 90 - Signaturparameter (Parameter)

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

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

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

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

104 2.1 Methode

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

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

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

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

```
122 <MethodeID> ::= "urn:" <NID> ":" <NSS>
123 <NID> ::= "pdfsigfilter"
124 <NSS> ::= <VENDOR> ":" <METHODE> ":" <VERSION>
125 <VENDOR> ::= "bka.gv.at" | 1*<URN chars>
126 <METHODE> ::= "text" | "binaer" | 1*<URN chars>
127 <VERSION> ::= "v" 1*<number> "." 1*<number> "." 1*<number>
128 <URN chars> ::= siehe <URN chars> in RFC 2141
129 <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.

130 **Beispiele:**

131 urn:pdfsigfilter:bka.gv.at:binaer:v1.0.0

132 urn:pdfsigfilter:bka.gv.at:text:v1.1.0

133 Der Abschnitt 4 dieser Spezifikation definiert Signaturmethoden im Detail.

134 **2.2 Parameter**

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

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

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

```

146 <PARAMETER_ID> ::= <SIGDEV_PROF> "@" [<PARAM_L1>] ["@" [<PARAM_L2>]]
147 <SIGDEV_PROF> ::= 1*<CHAR> | ""
148 <PARAM_L1> ::= 1*<CHAR>
149 <PARAM_L2> ::= 1*<CHAR>
150 <CHAR> ::= siehe <CHAR> in Abschnitt 6.1 in RFC 2234

```

<PARAMETER_ID>	Der Signaturparameter.
<SIGDEV_PROF>	Kennzeichnung des Signaturerstellungskomponente oder des konkret angewandten Signaturparameter-Profiles (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.

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.

192 4 Definierte Signaturmethoden

193 Die vorliegende Spezifikation definiert eine Reihe von Signaturmethoden, die wie folgt in
194 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

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

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

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

204 4.1 Textuelle Signatur, Version 1.0.0

205 4.1.1 Charakteristik

206 <i>Methoden-Kennzeichnung:</i>	urn:pdfsigfilter:bka.gv.at:text:v1.0.0
207 <i>Input-Datenstrom:</i>	das zu signierende PDF-Dokument 208 (binärer Datenstrom, application/pdf)
209 <i>Signierter Datenstrom:</i>	der aus dem PDF-Dokument extrahierte Text 210 (binärer Datenstrom, text/plain)
211 <i>Art der Signatur:</i>	XML-Signatur, Enveloping Signature
212 <i>Zulässige Signaturparameter:</i>	Default (MOA), Default (BKU), etsi-bka-1.0
213 <i>Anwendbarkeit:</i>	NICHT EMPFOHLEN 214 deprecated, wurde ersetzt durch 215 urn:pdfsigfilter:bka.gv.at:text:v1.1.0

216 4.1.2 Aufbereitung der zu signierenden Daten

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

219 Der Input-Datenstrom MUSS wie folgt behandelt werden:

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

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

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

261 4.1.3 XML-Signaturformat

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

265 Der Transformationspfad MUSS die folgenden Transformationen in dieser Reihenfolge
266 enthalten:

- 267 1. Base-64 Transformation der zu signierenden Daten (Algorithmus-Identifizier
268 <http://www.w3.org/2000/09/xmldsig#base64>)

269 Die zu erstellende XML-Signatur ist eine Enveloping Signature gem. [4], welche in Form eines
270 Datenobjekts die signierten Daten eingebettet enthält. Diese MÜSSEN Base-64 kodiert als
271 dsig:Object Element in die XML-Signatur eingebettet werden (näheres dazu siehe [4] und
272 [6]).

273 Die erstellte XML-Signatur folgt den Vorgaben des österreichischen E-Governments bzw. den
274 Vorgaben für XML-Signaturen aus der Spezifikation der österreichischen Bürgerkarte (siehe
275 [6]).

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

```
278 <dsig:Signature Id="signature-1161003152-26578093-24674"  
279 xmlns:dsig="http://www.w3.org/2000/09/xmldsig#">  
280   <dsig:SignedInfo>  
281     <dsig:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-  
282 20010315"/>  
283     <dsig:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#ecdsa-  
284 sha1"/>  
285     <dsig:Reference Id="signed-data-reference-0-1161003152-26578093-5873" URI="#signed-  
286 data-object-0-1161003152-26578093-8480">  
287       <dsig:Transforms>  
288         <dsig:Transform Algorithm="http://www.w3.org/2002/06/xmldsig-filter2">  
289           <xpf:XPath Filter="intersect" xmlns:xpf="http://www.w3.org/2002/06/xmldsig-  
290 filter2">id(&apos;signed-data-object-0-1161003152-26578093-  
291 8480&apos;)/node()/</xpf:XPath>  
292         </dsig:Transform>  
293         <dsig:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#base64"/>  
294       </dsig:Transforms>  
295       <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>  
296       <dsig:DigestValue>HASH-WERT DER 1. REFERENZ</dsig:DigestValue>  
297     </dsig:Reference>  
298     <dsig:Reference Id="etsi-data-reference-0-1161003152-26578093-26221"  
299 Type="http://uri.etsi.org/01903/v1.1.1#SignedProperties" URI="#etsi-data-object-0-  
300 1161003152-26578093-25255">  
301       <dsig:Transforms>  
302         <dsig:Transform Algorithm="http://www.w3.org/2002/06/xmldsig-filter2">  
303           <xpf:XPath Filter="intersect" xmlns:xpf="http://www.w3.org/2002/06/xmldsig-  
304 filter2">id(&apos;etsi-data-object-0-1161003152-26578093-  
305 25255&apos;)/node()/</xpf:XPath>  
306         </dsig:Transform>  
307       </dsig:Transforms>  
308       <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>  
309       <dsig:DigestValue>HASH-WERT DER 2. REFERENZ</dsig:DigestValue>  
310     </dsig:Reference>  
311   </dsig:SignedInfo>  
312   <dsig:SignatureValue>SIGNATURWERT</dsig:SignatureValue>  
313   <dsig:KeyInfo>  
314     <dsig:X509Data>  
315       <dsig:X509Certificate>ZERTIFIKAT</dsig:X509Certificate>  
316     </dsig:X509Data>  
317   </dsig:KeyInfo>  
318   <dsig:Object Id="signed-data-object-0-1161003152-26578093-8480">  
319     <sl:Base64Content>SIGNIERTE DATEN (BASE64)</sl:Base64Content>  
320   </dsig:Object>
```

```
321 <dsig:Object Id="etsi-data-object-0-1161003152-26578093-25255">
322   <etsi:QualifyingProperties Target="#signature-1161003152-26578093-24674"
323   xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
324   xmlns:etsi="http://uri.etsi.org/01903/v1.1.1#">
325     <etsi:SignedProperties>
326       <etsi:SignedSignatureProperties>
327         <etsi:SigningTime>SIGNATURZEITPUNKT</etsi:SigningTime>
328         <etsi:SigningCertificate>
329           <etsi:Cert>
330             <etsi:CertDigest>
331               <etsi:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
332               <etsi:DigestValue>HASHWERT DES SIGNATURZERTIFIKATES</etsi:DigestValue>
333             </etsi:CertDigest>
334             <etsi:IssuerSerial>
335               <dsig:X509IssuerName>AUSSTELLER DES ZERTIFIKATS</dsig:X509IssuerName>
336               <dsig:X509SerialNumber>SERIENNUMMER DES ZERTIFIKATS</dsig:X509SerialNumber>
337             </etsi:IssuerSerial>
338           </etsi:Cert>
339         </etsi:SigningCertificate>
340         <etsi:SignaturePolicyIdentifier>
341           <etsi:SignaturePolicyImplied/>
342         </etsi:SignaturePolicyIdentifier>
343       </etsi:SignedSignatureProperties>
344       <etsi:SignedDataObjectProperties>
345         <etsi:DataObjectFormat ObjectReference="#signed-data-reference-0-1161003152-
346 26578093-5873">
347           <etsi:MimeType>text/plain</etsi:MimeType>
348         </etsi:DataObjectFormat>
349       </etsi:SignedDataObjectProperties>
350     </etsi:SignedProperties>
351   </etsi:QualifyingProperties>
352 </dsig:Object>
353 </dsig:Signature>
```

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

358 4.1.4 Einbettung der Signatur in das PDF-Dokument

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

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

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

369 4.1.5 Anwendungshinweis zur Verifikation

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

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

377 Die Vorgehensweise der Verifikation im Überblick:

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

393 **4.2 Textuelle Signatur, Version 1.1.0**

394 **4.2.1 Charakteristik**

395 <i>Methoden-Kennzeichnung:</i>	urn:pdfsigfilter:bka.gv.at:text:v1.1.0
396 <i>Input-Datenstrom:</i>	das zu signierende PDF-Dokument 397 (binärer Datenstrom, application/pdf)
398 <i>Signierte Datenstrom:</i>	der aus dem PDF-Dokument extrahierte Text 399 (binärer Datenstrom, text/plain)
400 <i>Art der Signatur:</i>	XML-Signatur, Detached Signature
401 <i>Zulässige Signaturparameter:</i>	keine Einschränkung
402 <i>Anwendbarkeit:</i>	EMPFOHLEN

403 **4.2.2 Aufbereitung der zu signierenden Daten**

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

406 Der Input-Datenstrom MUSS wie folgt behandelt werden:

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

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

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

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

448 **4.2.3 XML-Signaturformat**

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

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

453 Die erstellte XML-Signatur folgt den Vorgaben des österreichischen E-Governments bzw. den
454 Vorgaben für XML-Signaturen aus der Spezifikation der österreichischen Bürgerkarte (siehe
455 [6]).

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

```
458 <dsig:Signature xmlns:dsig="http://www.w3.org/2000/09/xmldsig#" Id="signature-  
459 1201005938-3018328-13744">  
460   <dsig:SignedInfo>  
461     <dsig:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-  
462 20010315"/>  
463     <dsig:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#ecdsa-  
464 sha1"/>  
465     <dsig:Reference Id="signed-data-reference-0-1201005938-3018328-20478"  
466 URI="urn:Document">  
467       <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>  
468       <dsig:DigestValue>HASH-WERT DER 1. REFERENZ</dsig:DigestValue>  
469     </dsig:Reference>  
470     <dsig:Reference Id="etsi-data-reference-0-1201005938-3018328-327"  
471 Type="http://uri.etsi.org/01903/v1.1.1#SignedProperties"  
472 URI="#xmlns(etsi=http://uri.etsi.org/01903/v1.1.1%23)%20xpointer(id('etsi-data-object-  
473 0-1201005938-3018328-  
474 18413')/child::etsi:QualifyingProperties/child::etsi:SignedProperties)">  
475       <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>  
476       <dsig:DigestValue>HASH-WERT DER 2. REFERENZ</dsig:DigestValue>  
477     </dsig:Reference>  
478   </dsig:SignedInfo>  
479   <dsig:SignatureValue>SIGNATURWERT</dsig:SignatureValue>  
480   <dsig:KeyInfo>  
481     <dsig:X509Data>  
482       <dsig:X509Certificate>ZERTIFIKAT</dsig:X509Certificate>  
483     </dsig:X509Data>  
484   </dsig:KeyInfo>  
485   <dsig:Object Id="etsi-data-object-0-1201005938-3018328-18413">  
486     <etsi:QualifyingProperties xmlns:etsi="http://uri.etsi.org/01903/v1.1.1#"  
487 Target="#signature-1201005938-3018328-13744">  
488       <etsi:SignedProperties>  
489         <etsi:SignedSignatureProperties>  
490           <etsi:SigningTime>SIGNATURZEITPUNKT</etsi:SigningTime>  
491           <etsi:SigningCertificate>  
492             <etsi:Cert>  
493               <etsi:CertDigest>  
494                 <etsi:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>  
495                 <etsi:DigestValue>HASH-WERT DES ZERTIFIKATS</etsi:DigestValue>  
496               </etsi:CertDigest>  
497             <etsi:IssuerSerial>  
498               <dsig:X509IssuerName>AUSSTELLER DES ZERTIFIKATS</dsig:X509IssuerName>  
499               <dsig:X509SerialNumber>SERIENNUMMER DES ZERTIFIKATS</dsig:X509SerialNumber>  
500             </etsi:IssuerSerial>  
501           </etsi:Cert>  
502         </etsi:SigningCertificate>  
503         <etsi:SignaturePolicyIdentifier>  
504           <etsi:SignaturePolicyImplied/>  
505         </etsi:SignaturePolicyIdentifier>  
506       </etsi:SignedSignatureProperties>  
507       <etsi:SignedDataObjectProperties>  
508         <etsi:DataObjectFormat ObjectReference="#signed-data-reference-0-1201005938-  
509 3018328-20478">  
510           <etsi:MimeType>text/plain</etsi:MimeType>  
511         </etsi:DataObjectFormat>  
512       </etsi:SignedDataObjectProperties>  
513     </etsi:SignedProperties>  
514   </etsi:QualifyingProperties>  
515 </dsig:Object>  
516 </dsig:Signature>
```

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

521 **4.2.4 Einbettung der Signatur in das PDF-Dokument**

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

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

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

532 **4.2.5 Anwendungshinweis zur Verifikation**

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

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

540 Die Vorgehensweise der Verifikation im Überblick:

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

556 **4.3 Binäre Signatur, Version 1.0.0**

557 **4.3.1 Charakteristik**

558	<i>Methoden-Kennzeichnung:</i>	urn:pdfsigfilter:bka.gv.at:binaer:v1.0.0
559	<i>Input-Datenstrom:</i>	das zu signierende PDF-Dokument
560		(binärer Datenstrom, application/pdf)
561	<i>Signierter Datenstrom:</i>	das aufbereitete PDF-Dokument
562		(binärer Datenstrom, application/pdf)
563	<i>Art der Signatur:</i>	XML-Signatur, Enveloping Signature
564	<i>Zulässige Signaturparameter:</i>	Default (MOA), Default (BKU), etsi-bka-1.0
565	<i>Anwendbarkeit:</i>	NICHT EMPFOHLEN
566		deprecated, wurde ersetzt durch
567		urn:pdfsigfilter:bka.gv.at:binaer:v1.1.0

568 **4.3.2 Aufbereitung der zu signierenden Daten**

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

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

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

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

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

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

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

604 4.3.3 XML-Signaturformat

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

608 Der Transformationspfad MUSS die folgenden Transformationen in dieser Reihenfolge
609 enthalten:

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

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

617 Die erstellte XML-Signatur folgt den Vorgaben des österreichischen E-Governments bzw. den
618 Vorgaben für XML-Signaturen aus der Spezifikation der österreichischen Bürgerkarte (siehe
619 [6]).

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

```
622 <dsig:Signature Id="signature-1161003152-26578093-24674"  
623 xmlns:dsig="http://www.w3.org/2000/09/xmldsig#">  
624   <dsig:SignedInfo>  
625     <dsig:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-  
626     20010315"/>  
627     <dsig:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#ecdsa-  
628     sha1"/>  
629     <dsig:Reference Id="signed-data-reference-0-1161003152-26578093-5873" URI="#signed-  
630     data-object-0-1161003152-26578093-8480">  
631       <dsig:Transforms>  
632         <dsig:Transform Algorithm="http://www.w3.org/2002/06/xmldsig-filter2">  
633           <xpf:XPath Filter="intersect" xmlns:xpf="http://www.w3.org/2002/06/xmldsig-  
634     filter2">id(&apos;signed-data-object-0-1161003152-26578093-  
635     8480&apos;)/node()/</xpf:XPath>  
636         </dsig:Transform>  
637         <dsig:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#base64"/>  
638       </dsig:Transforms>  
639       <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>  
640       <dsig:DigestValue>HASH-WERT DER 1. REFERENZ</dsig:DigestValue>  
641     </dsig:Reference>  
642     <dsig:Reference Id="etsi-data-reference-0-1161003152-26578093-26221"  
643     Type="http://uri.etsi.org/01903/v1.1.1#SignedProperties" URI="#etsi-data-object-0-  
644     1161003152-26578093-25255">  
645       <dsig:Transforms>  
646         <dsig:Transform Algorithm="http://www.w3.org/2002/06/xmldsig-filter2">  
647           <xpf:XPath Filter="intersect" xmlns:xpf="http://www.w3.org/2002/06/xmldsig-  
648     filter2">id(&apos;etsi-data-object-0-1161003152-26578093-  
649     25255&apos;)/node()/</xpf:XPath>  
650         </dsig:Transform>  
651       </dsig:Transforms>
```

```
652 <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
653 <dsig:DigestValue>HASH-WERT DER 2. REFERENZ</dsig:DigestValue>
654 </dsig:Reference>
655 </dsig:SignedInfo>
656 <dsig:SignatureValue>SIGNATURWERT</dsig:SignatureValue>
657 <dsig:KeyInfo>
658 <dsig:X509Data>
659 <dsig:X509Certificate>ZERTIFIKAT</dsig:X509Certificate>
660 </dsig:X509Data>
661 </dsig:KeyInfo>
662 <dsig:Object Id="signed-data-object-0-1161003152-26578093-8480">
663 <sl:Base64Content>SIGNIERTE DATEN (BASE64)</sl:Base64Content>
664 </dsig:Object>
665 <dsig:Object Id="etsi-data-object-0-1161003152-26578093-25255">
666 <etsi:QualifyingProperties Target="#signature-1161003152-26578093-24674"
667 xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
668 xmlns:etsi="http://uri.etsi.org/01903/v1.1.1#">
669 <etsi:SignedProperties>
670 <etsi:SignedSignatureProperties>
671 <etsi:SigningTime>SIGNATURZEITPUNKT</etsi:SigningTime>
672 <etsi:SigningCertificate>
673 <etsi:Cert>
674 <etsi:CertDigest>
675 <etsi:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
676 <etsi:DigestValue>HASH-WERT DES SIGNATURZERTIFIKATS</etsi:DigestValue>
677 </etsi:CertDigest>
678 <etsi:IssuerSerial>
679 <dsig:X509IssuerName>AUSSTELLER DES ZERTIFIKATS</dsig:X509IssuerName>
680 <dsig:X509SerialNumber>SERIENNUMMER DES ZERTIFIKATS</dsig:X509SerialNumber>
681 </etsi:IssuerSerial>
682 </etsi:Cert>
683 </etsi:SigningCertificate>
684 <etsi:SignaturePolicyIdentifier>
685 <etsi:SignaturePolicyImplied/>
686 </etsi:SignaturePolicyIdentifier>
687 </etsi:SignedSignatureProperties>
688 <etsi:SignedDataObjectProperties>
689 <etsi:DataObjectFormat ObjectReference="#signed-data-reference-0-1161003152-
690 26578093-5873">
691 <etsi:MimeType>text/plain</etsi:MimeType>
692 </etsi:DataObjectFormat>
693 </etsi:SignedDataObjectProperties>
694 </etsi:SignedProperties>
695 </etsi:QualifyingProperties>
696 </dsig:Object>
697 </dsig:Signature>
```

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

702 **4.3.4 Einbettung der Signatur in das PDF-Dokument**

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

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

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

714 Der nachfolgende Abschnitt beschreibt das EGIZ-Dictionary im Detail.

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

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

726 **4.3.4.1 EGIZ-Dictionary**

727 Die Keys des EGIZ Dictionaries MÜSSEN, falls verwendet, in folgender Reihenfolge vorhanden
728 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	<p>Enthält aufsteigend sortierte Byte-Ranges, wodurch die signierten Bereiche des Dokumentes auf binärer Ebene identifiziert werden.</p> <p>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.</p> <p>Siehe Abschnitt 4.3.4.1.2.</p>
/replaces	MUSS	<p>Dieses Feld gibt an, welche Elemente der Signatur-Repräsentation nach der Signaturerstellung durch signaturspezifische Werte ersetzt werden müssen.</p> <p>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.</p> <p>Gibt es in einem Dokument n Byte-Ranges, so enthält das /replaces-Array (n-1) Einträge für die (n-1) Bereiche.</p> <p>Siehe Abschnitt 4.3.4.1.3.</p>
/encodings	MUSS	<p>Das Feld stellt ein PDF Array von PDF-Names dar, welche angeben, welches Encoding bei den Daten der Signaturattribute angewendet wurde.</p> <p>Die Reihenfolge der in diesem Array angegebenen Encodings MUSS mit der Reihenfolge des /replaces-Arrays korrespondieren.</p> <p>Siehe Abschnitt 4.3.4.1.4.</p>
/Cert	SOLL	<p>Dieses Array von PDF-Strings KANN zur Einbettung von Zertifikatsdaten (X509-Zertifikaten) verwendet werden. Es SOLL zumindest das Signaturzertifikat selbst, es KANN jedoch auch die gesamte Zertifikatskette im Rahmen dieses Arrays als PDF-Strings eingebettet werden.</p> <p>Siehe Abschnitt 4.3.4.1.5.</p>

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.

733 **4.3.4.1.1 /ID**

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

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

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

740 Zusammengefügt MÜSSEN die durch diese Byte-Ranges spezifizierten Strings wieder den
741 Signaturmethoden-String ergeben. Das Character Encoding des Kennzeichnungsstrings MUSS
742 8 bit WinAnsiEncoding sein. Zu beachten sind auch PDF String Escape Sequences (siehe
743 Kapitel „Strings in den Löchern“).

744 **4.3.4.1.2 /ByteRange**

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

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

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

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

761 **Beispiel:**

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

764 [0 100 110 90]

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

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

771 **Beispiel:**
 772 [...]

773 1 0 0 1 137.91 207 Tm

774 /F1 12 Tf

775 (123456789a) Tj

776 ET

777 BT

778 1 0 0 1 217.11 225 Tm

779 /F2 12 Tf

780 (123bcdefgh) Tj

781 1 0 0 1 217.11 213 Tm

782 [...]

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

788 **4.3.4.1.3 /replaces**

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

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

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

797 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).

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

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

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

812 **Beispiel:**

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

814 **4.3.4.1.4 /encodings**

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

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

822

823 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 <code>WinAnsiEncoding</code> (8bit) codiert. (siehe PDF Reference 1.4 [2], Appendix D).
/url	Die Zeichenkette wurde zuerst URL-kodiert (URL-encoded) danach mittels PDF <code>WinAnsiEncoding</code> (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.

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

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

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

835 **Beispiel:**

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

837 4.3.4.1.5 /Cert

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

844 4.3.4.1.6 Zeichenketten (Strings) als Wert

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

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

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

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

864 1. „\“ werden durch „\\“ ersetzt.

865 2. „(“ werden durch „\ („“ ersetzt.

866 3. „)“ werden durch „\)“ ersetzt.

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

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

874 **4.3.5 Anwendungshinweis zur Verifikation**

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

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

882 Die Vorgehensweise der Verifikation im Überblick:

883 1. Aus dem zu verifizierenden PDF-Dokument muss der zur Signatur gehörende Incremental
884 Update Block – beinhaltet das relevante EGIZ-Dictionary – extrahiert werden.

885 2. Über die im EGIZ-Dictionary eingetragenen Byte-Ranges werden die Aussparungen der
886 binären PDF-Signatur ermittelt.

887 3. Aus den ermittelten Aussparung werden die Werte extrahiert und gemäß den Vorgaben
888 dieser Art (Methode) von binären PDF-Signatur als Daten zur Rekonstruktion der XML-
889 Signatur interpretiert (d.h. Als Signaturattribute, wie Signaturwert, Signaturzeitpunkt, etc.,
890 bzw. als Signaturparameter sowie Kennzeichnung des angewandten Signaturparameter-
891 Profils). Diese Daten werden gem. den Vorgaben dieser Art (Methode) von binären PDF-
892 Signatur behandelt, d.h. zusammengeführt und kodiert, und interpretiert.

893 4. Der Wertebereich der Aussparungen wird gem. den Vorgaben dieser Art (Methode) von
894 binären PDF-Signatur mit den definierten Füllzeichen (Default-Werten) aufgefüllt werden.
895 Das danach resultierende PDF-Dokument repräsentiert das binär signierte PDF-
896 Dokument; dies entspricht dem signierten Datenstrom.

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

902 **4.4 Binäre Signatur, Version 1.1.0**

903 **4.4.1 Charakteristik**

904 <i>Methoden-Kennzeichnung:</i>	urn:pdfsigfilter:bka.gv.at:binaer:v1.1.0
905 <i>Input-Datenstrom:</i>	das zu signierende PDF-Dokument 906 (binärer Datenstrom, application/pdf)
907 <i>Signierte Datenstrom:</i>	das aufbereitet PDF-Dokument 908 (binärer Datenstrom, application/pdf)
909 <i>Art der Signatur:</i>	XML-Signatur, Detached Signature
910 <i>Zulässige Signaturparameter:</i>	keine Einschränkung
911 <i>Anwendbarkeit:</i>	EMPFOHLEN

912 **4.4.2 Aufbereitung der zu signierenden Daten**

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

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

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

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

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

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

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

946 4.4.3 XML-Signaturformat

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

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

951 Die erstellte XML-Signatur folgt den Vorgaben des österreichischen E-Governments bzw. den
952 Vorgaben für XML-Signaturen aus der Spezifikation der österreichischen Bürgerkarte (siehe
953 [6]).

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

```
956 <dsig:Signature xmlns:dsig="http://www.w3.org/2000/09/xmldsig#" Id="signature-  
957 1201006039-3118750-12731">  
958   <dsig:SignedInfo>  
959     <dsig:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-  
960 20010315"/>  
961     <dsig:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#ecdsa-  
962 sha1"/>  
963     <dsig:Reference Id="signed-data-reference-0-1201006039-3118750-13501"  
964 URI="urn:Document">  
965       <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>  
966       <dsig:DigestValue>HASH-WERT 1. REFERENZ</dsig:DigestValue>  
967     </dsig:Reference>  
968     <dsig:Reference Id="etsi-data-reference-0-1201006039-3118750-21710"  
969 Type="http://uri.etsi.org/01903/v1.1.1#SignedProperties"  
970 URI="#xmlns(etsi=http://uri.etsi.org/01903/v1.1.1%23)%20xpointer(id('etsi-data-object-  
971 0-1201006039-3118750-  
972 21127'))/child::etsi:QualifyingProperties/child::etsi:SignedProperties)">  
973       <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>  
974       <dsig:DigestValue>HASH-WERT 2. REFERENZ</dsig:DigestValue>  
975     </dsig:Reference>  
976   </dsig:SignedInfo>  
977   <dsig:SignatureValue>SIGNATURWERT</dsig:SignatureValue>  
978   <dsig:KeyInfo>  
979     <dsig:X509Data>  
980       <dsig:X509Certificate>ZERTIFIKAT</dsig:X509Certificate>  
981     </dsig:X509Data>  
982   </dsig:KeyInfo>  
983   <dsig:Object Id="etsi-data-object-0-1201006039-3118750-21127">  
984     <etsi:QualifyingProperties xmlns:etsi="http://uri.etsi.org/01903/v1.1.1#"   
985 Target="#signature-1201006039-3118750-12731">  
986       <etsi:SignedProperties>  
987         <etsi:SignedSignatureProperties>  
988           <etsi:SigningTime>SIGNATURZEITPUNKT</etsi:SigningTime>  
989           <etsi:SigningCertificate>  
990             <etsi:Cert>  
991               <etsi:CertDigest>  
992                 <etsi:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>  
993                 <etsi:DigestValue>HASH-WERT DES ZERTIFIKATS</etsi:DigestValue>  
994               </etsi:CertDigest>
```

```
995     <etsi:IssuerSerial>  
996     <dsig:X509IssuerName>AUSSTELLER DES ZERTIFIKATS</dsig:X509IssuerName>  
997     <dsig:X509SerialNumber>SERIENNUMMER DES ZERTIFIKATS</dsig:X509SerialNumber>  
998     </etsi:IssuerSerial>  
999     </etsi:Cert>  
1000  </etsi:SigningCertificate>  
1001  <etsi:SignaturePolicyIdentifier>  
1002    <etsi:SignaturePolicyImplied/>  
1003  </etsi:SignaturePolicyIdentifier>  
1004  </etsi:SignedSignatureProperties>  
1005  <etsi:SignedDataObjectProperties>  
1006    <etsi:DataObjectFormat ObjectReference="#signed-data-reference-0-1201006039-  
1007  3118750-13501">  
1008      <etsi:MimeType>application/pdf</etsi:MimeType>  
1009    </etsi:DataObjectFormat>  
1010  </etsi:SignedDataObjectProperties>  
1011  </etsi:SignedProperties>  
1012  </etsi:QualifyingProperties>  
1013  </dsig:Object>  
1014 </dsig:Signature>
```

1015 Dieses Beispiel enthält einige Besonderheiten der Signaturerstellungskomponente
1016 (Bürgerkartensoftware und Signaturerstellungseinheit), auf die in Verbindung mit
1017 Signaturparametern noch eingegangen wird. Aus Gründen der Übersichtlichkeit wurden
1018 variable Inhalte größtenteils durch verbale Umschreibungen ersetzt (eingerahmter Text).

1019 4.4.4 Einbettung der Signatur in das PDF-Dokument

1020 Die resultierende XML-Signatur MUSS letztlich in Form der in Abschnitt 3 definierten
1021 Repräsentation in das PDF-Dokument integriert werden.

1022 Die Einbettung der Signatur-Repräsentation (Signaturblock) im PDF-Dokument MUSS mit Hilfe
1023 eines Inkrementellen Update Blocks (Incremental Update Block, Abschnitt 3.4.5 in [2]) realisiert
1024 werden. Dieser Block MUSS folgende Struktur aufweisen:

- 1025 1. Dieser Incremental Update Block MUSS ein eigenes Dictionary, das EGIZ-Dictionary,
1026 enthalten. Dieses MUSS ein indirektes Objekt sein.
- 1027 2. Der gesamte sichtbare Signaturblock MUSS in ein XObject Form eingebettet sein. (siehe
1028 /SigXObject Key des EGIZ Dictionaries, Abschnitt 4.4.4.1)
- 1029 3. Das trailer-Dictionary des Incremental Update Blocks MUSS einen Key /EGIZSigDict
1030 enthalten. Wert dieses Keys MUSS eine indirekte Referenz auf das EGIZ-Dictionary sein.

1031 Der nachfolgende Abschnitt beschreibt das EGIZ-Dictionary im Detail.

1032 Als Spezifikum dieses Algorithmus MUSS die Signatur-Repräsentation bereits vor dem
1033 Signaturvorgang, im Zuge der Aufbereitung der zu signierenden Daten, in das PDF-Dokument
1034 eingebracht werden. Anstelle der zu diesem Zeitpunkt noch unbekanntenen Werte, wie bspw.
1035 Signaturattribute (das sind zum Beispiel Signaturwert, Signaturzeitpunkt, Angaben zum
1036 Signaturzertifikat, etc.), MÜSSEN die dafür vorgesehenen Wertebereiche mit semantisch
1037 wertfreien Füllzeichen aufgefüllt werden.

1038 Das mit diesem vorbereiteten aber leeren Signaturblock versehene PDF-Dokument wird in
1039 seiner binären Repräsentation elektronisch signiert. Nach dem Signaturprozess MÜSSEN die
1040 dabei ermittelten Werte (Signaturattribute) in die dafür vorgesehenen Wertebereiche der
1041 Signatur-Repräsentation (in den nachfolgenden Abschnitten auch als ‚Ausparung‘ bezeichnet)
1042 eingefüllt werden.

1043

1044 **4.4.4.1 EGIZ-Dictionary**

1045 Die Keys des EGIZ Dictionaries MÜSSEN, falls verwendet, in folgender Reihenfolge vorhanden
1046 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-Zertifikaten) 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.

1047 Sämtliche Werte, falls nicht anders spezifiziert, MÜSSEN direkte Objekte, also direkt im EGIZ-
1048 Dictionary eingebettet sein.

1049 Das EGIZ-Dictionary DARF noch weitere Elemente enthalten. Diese können dazu verwendet
1050 werden die Signatur mit zusätzlicher Information auszustatten.

1051 4.4.4.1.1 /ID

1052 Das /ID-Element beschreibt die Byte-Ranges (siehe Beschreibung /ByteRange) der
1053 Zeichenkette (String), die die angewendete Signaturmethode (Methode) identifiziert.

1054 Der so identifizierte String ist jener Teil des Content Streams der Signatur-Repräsentation,
1055 welcher den Text der Signaturmethode enthält. Dieser String unterliegt genauso den PDF
1056 Formatierungsregeln und wird daher im Allgemeinen in mehrere PDF-Strings gebrochen sein.

1057 Die /ID Byte-Ranges beschreiben den Inhalt dieser PDF-Strings des Content Streams.

1058 Zusammengefügt MÜSSEN die durch diese Byte-Ranges spezifizierten Strings wieder den
1059 Signaturmethoden-String ergeben. Das Character Encoding des Kennzeichnungsstrings MUSS
1060 8 bit WinAnsiEncoding sein. Zu beachten sind auch PDF String Escape Sequences (siehe
1061 Kapitel „Strings in den Löchern“).

1062 4.4.4.1.2 /ByteRange

1063 Statische Bereiche werden durch Byte-Ranges beschrieben. Byte-Ranges sind in Analogie zu
1064 Byte-Ranges in Adobe PDF-Signaturen definiert (siehe PDF Reference 1.6 [3], Kapitel 8.7):

- 1065 • Eine Byte-Range MUSS aus dem Zahlentupel Startoffset und Länge (in Bytes) bestehen.
- 1066 • Beide MÜSSEN positive Ganzzahlen sein, wobei der Startoffset auch 0 sein DARF.
- 1067 • Der Startoffset MUSS vom Anfang der PDF-Datei an gemessen werden.
- 1068 • Die Länge MUSS die Anzahl an Bytes ab dem Startoffset angeben, welche zur Byte-
1069 Range gehören.

1070 In einer binären Signatur MÜSSEN alle statischen Bereiche mittels Byte-Ranges identifiziert
1071 werden. Die variablen Bereiche zwischen den Byte-Ranges werden als Aussparungen
1072 bezeichnet, da die dadurch binär identifizierten Bereiche des PDF-Dokumentes nicht von der
1073 Signatur abgedeckt werden. In die Bereiche dieser Aussparungen MÜSSEN zum
1074 Signierzeitpunkt sogenannte Platzhalter-Zeichen eingefügt sein. Nach erfolgter Signatur
1075 MÜSSEN diese durch die resultierenden Werte der Signatur (zum Beispiel Signaturwert,
1076 Signaturzeitpunkt, etc.) ersetzt werden.

1077 Die Angaben von Byte-Ranges in den Elementen der binären Signatur MÜSSEN immer gemäß
1078 ihrer Startoffsets aufsteigend sortiert sein.

1079 **Beispiel:**

1080 Die Byte-Range (10, 5) beschreibt die Bytes an den Positionen 10, 11, 12, 13 und 14. Ein
1081 korrespondierendes /ByteRange-Array könnte folgendermaßen aussehen:

1082 [0 100 110 90]

1083 Dieses beschreibt zwei Byte-Ranges – von 0 bis 99 sowie von 110 bis 199 – und eine
1084 Aussparung – von 100 bis 109.

1085 Mit Hilfe dieser Byte-Range Angaben werden die signierten Bereiche des Dokuments auf Byte-
1086 Ebene identifiziert. Umgekehrt werden damit jene, nichtsignierten Aussparungen identifiziert, in
1087 die nach der Signatur die Signaturattribute eingebettet werden müssen. Das nachfolgende
1088 Beispiel soll dies illustrieren.

1089 **Beispiel:**

```
1090     [...]
1091     1 0 0 1 137.91 207 Tm
1092     /F1 12 Tf
1093     ([123456789a) Tj
1094     ET
1095     BT
1096     1 0 0 1 217.11 225 Tm
1097     /F2 12 Tf
1098     ([123bcdefgh) Tj
1099     1 0 0 1 217.11 213 Tm
1100     [...]
```

1101 Die nicht umrahmten Bereiche sind jene Bereiche, die über die Angabe von Byte-Ranges
1102 als die „signierten Bytes“ des Dokuments identifiziert werden würden. Die umrahmten
1103 Bereiche stellen hingegen Aussparungen dar, welche im signierten Dokument durch
1104 semantisch wertfreie Platzhalter ersetzt werden. Diese Struktur wird durch das
1105 /ByteRange-Element auf Byte-Ebene beschrieben.

1106 **4.4.4.1.3 /replaces**

1107 Das Feld stellt ein PDF Array von PDF-Names dar, welche angeben, welches Signaturattribut in
1108 den Aussparungen der binären Signatur – binär repräsentiert durch die inversen Byte-Ranges
1109 des /ByteRange-Elements – enthalten sind.

1110 Es MUSS so viele Elemente im /replaces Array geben wie es Aussparungen durch die
1111 getroffene Definition der Byte-Ranges gibt. Gibt es in einem Dokument n Byte-Ranges, so
1112 enthält das /replaces-Array (n-1) Einträge für die (n-1) Aussparungen.

1113 Jedes Element im /replaces Array MUSS ein gültiger PDF-Name sein, welcher den
1114 semantischen Wert (Typ) des Inhalts der Aussparung festlegt.

1115 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 EGZ-Dictionary (siehe Definition des /Cert-Array).
/dat	Zeichenkette, die den Signaturzeitpunkt repräsentiert/beschreibt (oder Teilfragment dessen).

PDF-Name	Semantischer Wert (Typ) / Signaturationtribut
/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).

1116 Unbekannte Typen MÜSSEN im Zuge der Verifikation der Signatur als nicht
1117 Spezifikationskonform zurückgewiesen werden.

1118 Signaturattribute, bzw. die sie repräsentierenden Zeichenketten, DÜRFEN aus Platzgründen
1119 auf mehrere, aufeinanderfolgende Aussparungen aufgeteilt werden. Der Inhalt von derart
1120 aufeinander folgenden Aussparungen gleichen Typs stellt zusammengefasst den Gesamtwert
1121 des betreffenden Signaturattributs dar. Um diesen Gesamtwert zu bilden MÜSSEN die
1122 Fragmente konkateniert werden.

1123 Das folgende Beispiel zeigt die Beschreibung der semantischen Inhalte von Aussparungen,
1124 wobei die erste Aussparung ein nicht näher spezifizierte Datenblock ist (Typ /nil), die zweite
1125 und die dritte Aussparung Fragmente der Zeichenkette des Signaturdatums enthalten (Typ
1126 /dat), die Aussparung vier bis sechs Fragmente der Zeichenkette des Signaturwertes
1127 enthalten (Typ /val), die siebte und achte Aussparung wiederum nicht näher spezifizierte
1128 Datenblöcke enthalten, die letzte Aussparung die die Signaturparameter beschreibende
1129 Zeichenkette repräsentiert (Typ /sid).

1130 **Beispiel:**

1131 `[/nil /dat /dat /val /val /val /nil /nil /sid]`

1132 4.4.4.1.4 /encodings

1133 Die Festlegung der Typen der in Aussparungen gefassten Werte durch das /replaces-Feld
1134 werden durch die analoge Festlegung von Kodierungsarten (Encoding) ergänzt. Das
1135 /encodings-Feld legt daher fest, auf welche Art die in den jeweiligen Aussparungen gefassten
1136 Daten (vor allem Zeichenketten) kodiert worden sind.

1137 Es MUSS für jeden Wert, der in einer Aussparung nach der Signatur eingebettet und im
1138 /replaces-Feld entsprechend festgelegt wurde, spezifiziert werden, auf welche Art dieser Wert
1139 kodiert ist.

1140 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 <code>WinAnsiEncoding</code> (8bit) codiert. (siehe PDF Reference 1.4 [2], Appendix D).
/url	Die Zeichenkette wurde zuerst URL-kodiert (URL-encoded) danach mittels PDF <code>WinAnsiEncoding</code> (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.

1141 Unbekannte Encoding-Arten MÜSSEN im Zuge der Verifikation der Signatur als nicht
1142 spezifikationskonform zurückgewiesen werden.

1143 Wird eine Zeichenkette (Signaturattribut) aus Platzgründen auf mehrere Aussparungen
1144 aufgeteilt (siehe Abschnitt 4.3.4.1.3), so MUSS für alle Fragmente dieser Zeichenkette das für
1145 das erste Fragment festgelegte Encoding (festgelegt durch das zur entsprechenden
1146 Aussparung korrespondierende Element des /encoding-Feldes) angenommen werden. Sind für
1147 die anderen Fragmente einer Zeichenkette davon abweichende Encodings festgelegt
1148 (festgelegt durch das zur entsprechenden Aussparung korrespondierende Element des
1149 /encoding-Feldes), so MÜSSEN diese ignoriert werden.

1150 Das nachfolgende Beispiel zeigt ein exemplarisches /encoding-Feld, im Einklang mit den zuvor,
1151 im Beispiel des Abschnitts 4.3.4.1.3, definierten Aussparungen.

1152 **Beispiel:**

1153 `[/nil /f16 /nil /nil /win /win /url]`

1154 4.4.4.1.5 /Cert

1155 Dieses Element wurde in Synergie mit dem /Cert Element von Adobe PDF Signaturen
1156 definiert (siehe PDF Reference 1.6 [3], Tabelle 8.98). Es gilt die Einschränkung, dass, falls
1157 vorhanden, der Wert immer ein Feld von direkten Literal-String-Objekten sein MUSS. Jeder
1158 String MUSS ein Base64-codiertes Zertifikat im PEM Format enthalten. Das erste Zertifikat
1159 MUSS das Signaturzertifikat sein. Die restlichen Zertifikate stellen die Zertifikatskette zu einem
1160 vertrauenswürdigen Wurzelzertifikat dar und sind OPTIONAL.

1161 4.4.4.1.6 Zeichenketten (Strings) als Wert

1162 Layoutbedingt wird eine längere Zeichenkette der Signatur-Repräsentation auf mehrere
1163 Aussparungen aufgeteilt. Umgekehrt muss in der visuellen Darstellung der Signatur-
1164 Repräsentation (Signaturblock) im PDF selbst genügend Raum geschaffen werden, um den
1165 gesamte Zeichenkette aufnehmen zu können. Dies bedeutet, dass der theoretisch mögliche
1166 Platz in Aussparungen nicht voll ausgefüllt werden kann, da diese Zeichenkette visuell, im
1167 gewählten Layout des Signaturblocks keinen Platz mehr findet (Darstellung würde bspw. Über
1168 den Papierrand reichen).

1169 Um die in einer Aussparung gefassten Zeichen/Werte vom ungenutzten Raum der
1170 Aussparungen unterscheiden zu können, MÜSSEN alle nicht genutzten Bereiche der
1171 Aussparung mit dem NULL-Byte (numerisch 0) befüllt werden. Das NULL-Byte DARF NICHT in
1172 einzusetzenden Werten vorkommen.

1173 Vor dem Einsetzen der Werte (kodierte Zeichenketten) in die Bereiche der Aussparungen
1174 MÜSSEN, unabhängig der verwendeten und im Feld /encoding festgelegten Kodierung,
1175 allfällige im einzusetzenden Wert (Zeichenkette) PDF-Steuerzeichen maskiert werden.

1176 Die zu maskierenden PDF-Steuerzeichen sind der Backslash („\“) sowie die linke und rechte
1177 Rundklammer („(,“ und „),“). Es MUSS jedes im einzusetzenden Wert (Zeichenkette)
1178 vorkommende PDF-Steuerzeichen durch einen Backslash eingeleitet (escaped) werden. Es
1179 MUSS daher vor dem Einsetzen der Zeichenkette/Werte folgende Ersetzung vorgenommen
1180 werden:

- 1181 1. „\“ werden durch „\\“ ersetzt.
- 1182 2. „(“ werden durch „\ („,“ ersetzt.
- 1183 3. „),“ werden durch „)\“ ersetzt.

1184 Bei der Rekonstruktion der Werte im Zuge der Verifikation MUSS diese Transformation in
1185 umgekehrter Reihenfolge durchgeführt und somit rückgängig gemacht werden.

1186 Hinweis: Es MUSS weiters darauf geachtet werden, dass diese ein derart ersetztes PDF-
1187 Steuerzeichen – die sogenannte „escape-sequence“ – niemals geteilt wird. Ein alleine
1188 stehender „\“ vor dem Ende des Bereichs einer Aussparung würde den Abschluss des Bereichs
1189 der Aussparung (End-Delimiter, im PDF mit „)“ dargestellt) „maskieren“ und damit das
1190 Dokument unbrauchbar machen.

1191 **4.4.5 Anwendungshinweis zur Verifikation**

1192 Zur Verifikation von derart signierten Dokumenten MUSS reziprok zu der in dieser Spezifikation
1193 festgelegten Vorgehensweise verfahren werden. Zusätzlich werden die folgenden
1194 Anwendungshinweise gegeben.

1195 Gegeben sei ein unter Anwendung der hier spezifizierten Signaturmethode textuell signiertes
1196 PDF-Dokument. Die Applikation MUSS aus der in der Signatur-Repräsentation enthaltenen
1197 Methoden-Kennung das korrekte Signaturverfahren bestimmen und somit das adäquate
1198 Verifikationsverfahren anwenden.

1199 Die Vorgehensweise der Verifikation im Überblick:

- 1200 1. Aus dem zu verifizierenden PDF-Dokument muss der zur Signatur gehörende Incremental
1201 Update Block – beinhaltet das relevante EGIZ-Dictionary – extrahiert werden.
- 1202 2. Über die im EGIZ-Dictionary eingetragenen Byte-Ranges werden die Aussparungen der
1203 binären PDF-Signatur ermittelt.
- 1204 3. Aus den ermittelten Aussparung werden die Werte extrahiert und gemäß den Vorgaben
1205 dieser Art (Methode) von binären PDF-Signatur als Daten zur Rekonstruktion der XML-
1206 Signatur interpretiert (d.h. Als Signaturattribute, wie Signaturwert, Signaturzeitpunkt, etc.,
1207 bzw. als Signaturparameter sowie Kennzeichnung des angewandten Signaturparameter-
1208 Profils). Diese Daten werden gem. den Vorgaben dieser Art (Methode) von binären PDF-
1209 Signatur behandelt, d.h. zusammengeführt und kodiert, und interpretiert.
- 1210 4. Der Wertebereich der Aussparungen wird gem. den Vorgaben dieser Art (Methode) von
1211 binären PDF-Signatur mit den definierten Füllzeichen (Default-Werten) aufgefüllt werden.
1212 Das danach resultierende PDF-Dokument repräsentiert das binär signierte PDF-
1213 Dokument; dies entspricht dem signierten Datenstrom.
- 1214 5. Die dem signierten PDF-Dokument hinterlegte XML-Signatur wird anhand der aus dem
1215 EGIZ-Dictionary gewonnenen Daten (siehe vorherigen Schritt) unter Berücksichtigung des
1216 jeweiligen Signaturparameter-Profils, bzw. unter Anwendung des damit festgelegten XML-
1217 Signaturlayouts, rekonstruiert.
- 1218 6. Die rekonstruierte XML-Signatur wird verifiziert.

1219 5 Definierte Signaturparameter

1220 Die Signaturparameter nehmen auf Spezialitäten von Signaturerstellungskomponenten
1221 Rücksicht. Vor allem werden damit das Layout und die variablen Elemente der damit erstellten
1222 XML-Signaturen im Rahmen der Signatur-Repräsentation (Signaturblock) zum Ausdruck
1223 gebracht. Dies ist besonders für den Fall der Signaturrekonstruktion auf Basis eines
1224 Papierausdruckes erforderlich (siehe auch allgemeine Erläuterung in Abschnitt 2.2).

1225 Die vorliegende Spezifikation berücksichtigt Spezialitäten einiger Signaturerstellungskomponenten und enthält daher die Definition der folgenden Signaturparameter-Profile, die wie
1226 folgt in Implementierungen unterstützt werden MÜSSEN:
1227

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

1228 Eine spezifikationskonforme Umsetzung MUSS die in der obigen Tabelle definierten
1229 Signaturparameterprofile, gemäß den definierten Prioritäten, implementieren.

1230 Jede Implementierung MUSS für die damit erzeugbaren Signaturparameterprofile sowohl die
1231 Signaturerstellung als auch die Signaturverifikation realisieren.

1232 5.1 Default Signaturparameter-Profil

1233 5.1.1 Charakteristik

1234 *Parameter-Kennzeichnung:* keine Kennzeichnung; es werden keine Parameter
1235 angeführt

1236 *Signaturerstellungskomponente:* für alle Signaturerstellungskomponenten
1237 gem. Spezifikation MOA-SS [7]

1238 *Einschränkungen bzgl. Signaturmethoden:* keine, kann mit allen Signaturmethoden verwendet werden
1239

1240 *Anwendbarkeit:* EMPFOHLEN

1241 5.1.2 Signaturparameter

1242 Für dieses Signaturparameter-Profil DÜRFEN Signaturparameter NICHT verwendet werden.

1243 5.1.3 Signaturlayout

1244 Werden keine Signaturparameter angegeben, so MUSS die erstellte Signatur den Vorgaben
1245 einer MOA-SS Signatur [7] folgen und DARF NICHT variable (zeitabhängige oder zufällige)
1246 Werte enthalten.

1247 Eine XML-Signatur, die diesem Parameter-Profil entspricht, MUSS dem folgenden Layout
1248 entsprechen:

```
1249 <dsig:Signature Id="signature-1-1" xmlns:dsig="http://www.w3.org/2000/09/xmldsig#">
1250   <dsig:SignedInfo xmlns:dsig="http://www.w3.org/2000/09/xmldsig#">
1251     <dsig:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-
1252       20010315"/>
1253     <dsig:SignatureMethod Algorithm="ALGORITHM"/>
1254     <dsig:Reference Id="reference-1-1" URI="REFERENCE">
1255       TRANSFORMS
1256       <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
1257       <dsig:DigestValue>DIGESTVALUESIGNEDDATA</dsig:DigestValue>
1258     </dsig:Reference>
1259     <dsig:Reference Type="http://uri.etsi.org/01903/v1.1.1#SignedProperties"
1260       URI="#xmlns(etsi=http://uri.etsi.org/01903/v1.1.1%23)%20xpointer(id('etsi-signed-1-
1261       1')/child:etsi:QualifyingProperties/child:etsi:SignedProperties)">
1262       <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
1263       <dsig:DigestValue>DIGESTVALUESIGNEDPROPERTIES</dsig:DigestValue>
1264     </dsig:Reference>
1265   </dsig:SignedInfo>
1266   <dsig:SignatureValue>SIGNATUREVALUE</dsig:SignatureValue>
1267   <dsig:KeyInfo>
1268     <dsig:X509Data>
1269       <dsig:X509Certificate>X509CERTIFICATE</dsig:X509Certificate>
1270     </dsig:X509Data>
1271   </dsig:KeyInfo>
1272   DSIGOBJECT
1273   <dsig:Object Id="etsi-signed-1-1">
1274     <etsi:QualifyingProperties Target="#signature-1-1"
1275       xmlns:etsi="http://uri.etsi.org/01903/v1.1.1#">
1276       <etsi:SignedProperties xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
1277         xmlns:etsi="http://uri.etsi.org/01903/v1.1.1#">
1278         <etsi:SignedSignatureProperties>
1279           <etsi:SigningTime>SIGNINGTIME</etsi:SigningTime>
1280           <etsi:SigningCertificate>
1281             <etsi:Cert>
1282               <etsi:CertDigest>
1283                 <etsi:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
1284                 <etsi:DigestValue>DIGESTVALUEX509CERTIFICATE</etsi:DigestValue>
1285               </etsi:CertDigest>
1286               <etsi:IssuerSerial>
1287                 <dsig:X509IssuerName>X509ISSUERNAME</dsig:X509IssuerName>
1288                 <dsig:X509SerialNumber>X509SERIALNUMBER</dsig:X509SerialNumber>
1289               </etsi:IssuerSerial>
1290             </etsi:Cert>
1291           </etsi:SigningCertificate>
1292           <etsi:SignaturePolicyIdentifier>
1293             <etsi:SignaturePolicyImplied/>
1294           </etsi:SignaturePolicyIdentifier>
1295         </etsi:SignedSignatureProperties>
```

```
1296     <etsi:SignedDataObjectProperties>  
1297         <etsi:DataObjectFormat ObjectReference="#reference-1-1">  
1298             <etsi:MimeType>MIMETYPE</etsi:MimeType>  
1299         </etsi:DataObjectFormat>  
1300     </etsi:SignedDataObjectProperties>  
1301 </etsi:SignedProperties>  
1302 </etsi:QualifyingProperties>  
1303 </dsig:Object>  
1304 </dsig:Signature>
```

1305 Die vom Layout vorgesehenen Variabilitäten werden in Großbuchstaben und umrandet
1306 ausgewiesen. Die nachfolgenden Abschnitte definieren diese im Detail.

1307 Dieses Profil legt durch das damit festgelegte XML-Signaturlayout die folgenden wesentlichen
1308 XML-Signatur-Eigenschaften implizit fest:

- 1309 1. Kanonisierungsmethode:
1310 <http://www.w3.org/TR/2001/REC-xml-c14n-20010315>
- 1311 2. Digest-Methode: <http://www.w3.org/2000/09/xmldsig#sha1>
- 1312 3. Qualifying Properties: nach ETSI (<http://uri.etsi.org/01903/v1.1.1#>)

1313 5.1.3.1 ALGORITHM

1314 Legt den Signaturalgorithmus fest. Es MUSS einer der folgenden Signaturalgorithmen
1315 verwendet werden, in Abhängigkeit der Eigenschaften des Signatur-Schlüssels:

- 1316 • für ECDSA-Schlüssel: <http://www.w3.org/2000/09/xmldsig#rsa-sha1>
- 1317 • für RSA-Schlüssel: <http://www.w3.org/2000/09/xmldsig#rsa-sha1>

1318 Der zum Signaturschlüssel passende Identifikator (URI) MUSS im XML-Signaturlayout anstelle
1319 der Variable **ALGORITHM** eingesetzt werden.

1320 5.1.3.2 REFERENCE

1321 Wird eine Signatur-Methode angewandt, die eine enveloping XML-Signatur erzeugt, so MUSS
1322 das folgende Fragment anstelle der Variable **REFERENCE** im XML-Signaturlayout eingesetzt
1323 werden:

```
1324     #xmlns(etsi=http://uri.etsi.org/01903/v1.1.1%23)%20xpointer(id('ets  
1325 i-signed-1-1')/child::etsi:QualifyingProperties/child::etsi:Sig  
1326 nedProperties)
```

1327 Wird eine Signatur-Methode angewandt, die eine detached XML-Signatur erzeugt, so MUSS
1328 das folgende Fragment anstelle der Variable **REFERENCE** im XML-Signaturlayout eingesetzt
1329 werden:

```
1330     urn:Document
```

1331 5.1.3.3 TRANSFORMS

1332 Wird eine Signatur-Methode angewandt, die eine enveloping XML-Signatur erzeugt, so MUSS
1333 das folgende Fragment anstelle der Variable **TRANSFORMS** im XML-Signaturlayout eingesetzt
1334 werden:

```
1335     <dsig:Transforms>  
1336         <dsig:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#base64"/>  
1337     </dsig:Transforms>
```

1338 Wird eine Signatur-Methode angewandt, die eine detached XML-Signatur erwirkt, so darf kein
1339 Transformationspfad eingefügt werden. In diesem Fall DARF ein Transformationspfad NICHT
1340 im XML-Signaturlayout eingesetzt werden. Die Variable **TRANSFORMS** im XML-Signaturlayout
1341 MUSS ersatzlos entfernt werden.

1342 **5.1.3.4 DIGESTVALUESIGNEDDATA**

1343 Anstelle der Variable `DIGESTVALUESIGNEDDATA` MUSS im XML-Signaturlayout der Hash-
1344 Wert der ersten XML-Signaturreferenz eingesetzt werden.

1345 **5.1.3.5 DIGESTVALUESIGNEDPROPERTIES**

1346 Anstelle der Variable `DIGESTVALUESIGNEDPROPERTIES` MUSS im XML-Signaturlayout der
1347 Hash-Wert der zweiten XML-Signaturreferenz eingesetzt werden.

1348 **5.1.3.6 SIGNATUREVALUE**

1349 Anstelle der Variable `SIGNATUREVALUE` MUSS im XML-Signaturlayout der Signaturwert
1350 eingesetzt werden.

1351 **5.1.3.7 X509CERTIFICATE**

1352 Anstelle der Variable `X509CERTIFICATE` MUSS im XML-Signaturlayout das Base64-kodierte
1353 Signaturzertifikat (X509-Zertifikat, kodiert gem. Abschnitt 4.4.4 in [4]) eingesetzt werden.

1354 **5.1.3.8 DSIGOBJECT**

1355 Wird eine Signatur-Methode angewandt, die eine enveloping XML-Signatur erzeugt, so MUSS
1356 das folgende Fragment anstelle der Variable `DSIGOBJECT` im XML-Signaturlayout eingesetzt
1357 werden:

```
1358     <dsig:Object Id="signed-data-1-1-1">  
1359         <Base64Content>BASE64CONTENT</Base64Content>  
1360     </dsig:Object>
```

1361 In diesem Fragment MUSS anstelle der Variable `BASE64CONTENT` der Base64-kodierte, zu
1362 signierende Datenstrom (gem. angewandter Signaturmethode) eingesetzt werden.

1363 Wird eine Signatur-Methode angewandt, die eine detached XML-Signatur erzeugt, so ist kein
1364 XML-Signaturobjekt für die signierten Daten notwendig. Die Variable `DSIGOBJECT` im XML-
1365 Signaturlayout MUSS ersatzlos entfernt werden.

1366 **5.1.3.9 SIGNINGTIME**

1367 Anstelle der Variable `SIGNINGTIME` MUSS der Signaturzeitpunkt gem. [5] eingesetzt werden.

1368 **5.1.3.10 DIGESTVALUEX509CERTIFICATE**

1369 Anstelle der Variable `DIGESTVALUEX509CERTIFICATE` MUSS der Hash-Wert des
1370 Signaturzertifikates (lt. [5]) eingesetzt werden.

1371 **5.1.3.11 X509ISSUERNAME**

1372 Anstelle der Variable `X509ISSUERNAME` MUSS der Name des Ausstellers des
1373 Signaturzertifikates (lt. [5]) eingesetzt werden.

1374 **5.1.3.12 X509SERIALNUMBER**

1375 Anstelle der Variable `X509SERIALNUMBER` MUSS die Seriennummer des Signaturzertifikates
1376 (lt. [5]) eingesetzt werden.

1377 **5.1.3.13 MIMETYPE**

1378 Anstelle der Variable `MIMETYPE` MUSS der MIME-Type der zu signierenden Daten eingesetzt
1379 werden. Der MIME-Type ist von der angewandten Signaturmethode abhängig und MUSS im
1380 Zuge der Definition der Signaturmethode festgelegt werden.

1381 **5.2 Default Signaturparameter-Profil für BKU**

1382 **5.2.1 Charakteristik**

1383	<i>Parameter-Kennzeichnung:</i>	keine Kennzeichnung; es werden zwar explizite
1384		Signaturparameter eingeführt, dieses Profil hat jedoch
1385		keine eigene Kennzeichnung. Es werden nur die Parameter
1386		in der Signatur-Repräsentation (Signaturblock) eingefügt.
1387	<i>Signaturerstellungskomponente:</i>	dieses Profil MUSS für die Generallizenz der
1388		Bürgerkartenumgebung (IT-Solution trustDesk-Basic), bis
1389		zur Version 2.7.4. angewendet werden.
1390	<i>Einschränkungen bzgl. Signatur-</i>	
1391	<i>methoden:</i>	dieses Profil MUSS mit Signaturmethoden verwendet
1392		werden, die eine enveloping Signatur erzeugen; diese sind:
1393		urn:pdfsigfilter:bka.gv.at:binaer:v1.0.0 und
1394		urn:pdfsigfilter:bka.gv.at:text:v1.0.0
1395	<i>Verwendbarkeit:</i>	Dieses Signaturparameter-Profil DARF NICHT mehr zur
1396		Anwendung gelangen (deprecated). Es wurde durch das
1397		ets-bka-1.0 Signaturparameter-Profil ersetzt.

1398 **5.2.2 Signaturparameter**

1399 Für dieses Signaturparameter-Profil wurden Signaturparameter definiert. Diese MÜSSEN ohne
1400 vorangestellte Kennzeichnung in die Signatur-Repräsentation eingefügt werden.

1401 Für die Formulierung der Signaturparameter MUSS die Definition gem. Abschnitt 2.2. Es gilt bei
1402 diesem Signaturparameter-Profil jedoch die Maßgabe, dass das Feld <SIGDEV_PROF> leer
1403 bleiben MUSS.

1404 Die Signaturparameter dieses Profils repräsentieren 5 Einzelwerte. Dabei MUSS als Parameter
1405 Teil 1 (Feld PARAM_L1 der Signaturparameter, siehe 2.2) als konstanter Präfix für alle 5
1406 Einzelwerte herangezogen werden. Der Parameter Teil 2 (Feld PARAM_L2 der
1407 Signaturparameter, siehe 2.2) enthält selbst 5 Einzelwerte, die jeweils mit einem Bindestrich
1408 separiert werden müssen. Es gilt für Parameter Teil 2 folgende Ausformung (in Ergänzung zur
1409 Definition aus Abschnitt 2.2):

```
1410 <PARAM_L2> ::= <WERT_1>"-"<WERT_2>"-"<WERT_3>"-"<WERT_4>"-"<WERT_5>  
1411 <WERT_1> ::= 1*<CHAR>  
1412 <WERT_2> ::= 1*<CHAR>  
1413 <WERT_3> ::= 1*<CHAR>  
1414 <WERT_4> ::= 1*<CHAR>  
1415 <WERT_5> ::= 1*<CHAR>
```

1416 Daraus MÜSSEN folgende Einzelwerte gebildet werden:

```
1417 <ParamSigID> ::= <PARAM_L1>"-"<WERT_1>  
1418 <ParamSigDataRef> ::= "0-"<PARAM_L1>"-"<WERT_2>  
1419 <ParamSigDataObjURI> ::= "0-"<PARAM_L1>"-"<WERT_3>  
1420 <ParamEtsiDataRef> ::= "0-"<PARAM_L1>"-"<WERT_4>  
1421 <ParamEtsiDataObjURI> ::= "0-"<PARAM_L1>"-"<WERT_5>
```

1422 Dieser Werte werden mehrfach im Signaturlayout referenziert und verwendet. Wird eine
1423 Signatur-Methode angewandt, die eine detached XML-Signatur erzeugt, so wird der Wert
1424 <ParamSigDataObjURI> nicht benötigt. Die Signaturparameter lassen die Bildung dieses
1425 Wertes jedoch dennoch zu.

1426 Die Verwendung dieser Einzelwerte wird in Abschnitt 5.2.3 festgelegt.

1427 Nachfolgend ein Beispiel zur Bildung der Einzelwerte auf Basis übergebener Signaturparameter
1428 (spezifisch für das vorliegende Signaturparameter-Profil):

1429 Signaturparameter lt. Signaturrepräsentation:

1430 @1200412799-27800484@23524-22018-26789-24095-30271

1431 Daraus ergeben sich:

1432 <ParamSigID> = 1200412799-27800484-23524
1433 <ParamSigDataRef> = 0-1200412799-27800484-22018
1434 <ParamSigDataObjURI> = 0-1200412799-27800484-26789
1435 <ParamEtsiDataRef> = 0-1200412799-27800484-24095
1436 <ParamEtsiDataObjURI> = 0-1200412799-27800484-30271

1437 5.2.3 Signaturlayout

1438 Die Signaturen einer Bürgerkartenumgebung enthalten zeitabhängige Varianzen. Diese sind vor
1439 allem eine Reihe von XML-Attributen (ID-Attribute) die zur Referenzierung von XML-
1440 Elementen/-Knoten herangezogen werden. Diese variablen Attribute MÜSSEN in Form der
1441 Signaturparameter innerhalb der Signatur-Repräsentation verwaltet und bei der Rekonstruktion
1442 der Signatur entsprechend berücksichtigt werden.

1443 Eine XML-Signatur, die diesem Parameter-Profil entspricht, MUSS dem folgenden Layout
1444 entsprechen:

```
1445 <dsig:Signature xmlns:dsig="http://www.w3.org/2000/09/xmldsig#" Id="signature-SIGID">  
1446   <dsig:SignedInfo>  
1447     <dsig:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-  
1448 20010315"/>  
1449     <dsig:SignatureMethod Algorithm="ALGORITHM"/>  
1450     <dsig:Reference Id="signed-data-reference-SIGDATAREF" URI="#signed-data-object-  
1451 SIGDATAOBJURI">  
1452       <dsig:Transforms>  
1453         <dsig:Transform Algorithm="http://www.w3.org/2002/06/xmldsig-filter2">  
1454           <xpf:XPath xmlns:xpf="http://www.w3.org/2002/06/xmldsig-filter2"  
1455 Filter="intersect">id('signed-data-object-SIGDATAOBJURI')/node()</xpf:XPath>  
1456           </dsig:Transform>  
1457           <dsig:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#base64"/>  
1458           </dsig:Transform>  
1459         <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>  
1460         <dsig:DigestValue>DIGESTVALUESIGNEDDATA</dsig:DigestValue>  
1461       </dsig:Reference>  
1462       <dsig:Reference Id="etsi-data-reference-ETSIDATAREF"  
1463 Type="http://uri.etsi.org/01903/v1.1.1#SignedProperties" URI="#etsi-data-object-  
1464 ETSIDATAOBJURI">  
1465         <dsig:Transforms>  
1466           <dsig:Transform Algorithm="http://www.w3.org/2002/06/xmldsig-filter2">  
1467             <xpf:XPath xmlns:xpf="http://www.w3.org/2002/06/xmldsig-filter2"  
1468 Filter="intersect">id('etsi-data-object-ETSIDATAOBJURI')/node()</xpf:XPath>  
1469             </dsig:Transform>  
1470           </dsig:Transforms>  
1471           <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>  
1472           <dsig:DigestValue>DIGESTVALUESIGNEDPROPERTIES</dsig:DigestValue>  
1473         </dsig:Reference>  
1474       </dsig:SignedInfo>  
1475       <dsig:SignatureValue>SIGNATUREVALUE</dsig:SignatureValue>  
1476       <dsig:KeyInfo>  
1477         <dsig:X509Data>  
1478           <dsig:X509Certificate>X509CERTIFICATE</dsig:X509Certificate>  
1479         </dsig:X509Data>  
1480       </dsig:KeyInfo>  
1481       <dsig:Object Id="signed-data-object-SIGDATAOBJURI">  
1482         <sl:Base64Content>BASE64CONTENT</sl:Base64Content>  
1483       </dsig:Object>
```

```
1484 <dsig:Object Id="etsi-data-object-ETSIDATAOBJURI">
1485   <etsi:QualifyingProperties xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
1486   xmlns:etsi="http://uri.etsi.org/01903/v1.1.1#" Target="#signature-SIGID">
1487     <etsi:SignedProperties>
1488       <etsi:SignedSignatureProperties>
1489         <etsi:SigningTime>SIGNINGTIME</etsi:SigningTime>
1490         <etsi:SigningCertificate>
1491           <etsi:Cert>
1492             <etsi:CertDigest>
1493               <etsi:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
1494               <etsi:DigestValue>DIGESTVALUEX509CERTIFICATE</etsi:DigestValue>
1495             </etsi:CertDigest>
1496             <etsi:IssuerSerial>
1497               <dsig:X509IssuerName>X509ISSUERNAME</dsig:X509IssuerName>
1498               <dsig:X509SerialNumber>X509SERIALNUMBER</dsig:X509SerialNumber>
1499             </etsi:IssuerSerial>
1500           </etsi:Cert>
1501         </etsi:SigningCertificate>
1502         <etsi:SignaturePolicyIdentifier>
1503           <etsi:SignaturePolicyImplied/>
1504         </etsi:SignaturePolicyIdentifier>
1505       </etsi:SignedSignatureProperties>
1506     <etsi:SignedDataObjectProperties>
1507       <etsi:DataObjectFormat ObjectReference="#signed-data-reference-SIGDATAREF">
1508         <etsi:MimeType>text/plain</etsi:MimeType>
1509       </etsi:DataObjectFormat>
1510     </etsi:SignedDataObjectProperties>
1511   </etsi:SignedProperties>
1512 </etsi:QualifyingProperties>
1513 </dsig:Object>
1514 </dsig:Signature>
```

1515 Die vom Layout vorgesehenen Variabilitäten (Variablen) werden in Großbuchstaben und
1516 umrandet ausgewiesen. Die nachfolgenden Abschnitte definieren diese im Detail.

1517 Dieses Profil legt durch das damit festgelegte XML-Signaturlayout die folgenden wesentlichen
1518 XML-Signatur-Eigenschaften implizit fest:

- 1519 1. Kanonisierungsmethode: <http://www.w3.org/TR/2001/REC-xml-c14n-20010315>
- 1520 2. Digest-Methode: <http://www.w3.org/2000/09/xmldsig#sha1>
- 1521 3. Qualifying Properties: nach ETSI (<http://uri.etsi.org/01903/v1.1.1#>)

1522 5.2.3.1 SIGID

1523 Anstelle der Variable **SIGID** MUSS im XML-Signaturlayout der aus den Signaturparametern
1524 zusammengesetzte Wert <ParamSigID> eingesetzt werden.

1525 5.2.3.2 ALGORITHM

1526 Legt den Signaturalgorithmus fest. Es MUSS einer der folgenden Signaturalgorithmen
1527 verwendet werden, in Abhängigkeit der Eigenschaften des Signatur-Schlüssels:

- 1528 • für ECDSA-Schlüssel: <http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha1>
- 1529 • für RSA-Schlüssel: <http://www.w3.org/2000/09/xmldsig#rsa-sha1>

1530 Der zum Signaturschlüssel passende Identifikator (URI) MUSS im XML-Signaturlayout anstelle
1531 der Variable **ALGORITHM** eingesetzt werden.

1532 5.2.3.3 SIGDATAREF

1533 Anstelle der Variable **SIGDATAREF** MUSS im XML-Signaturlayout der aus den
1534 Signaturparametern zusammengesetzte Wert <ParamSigDataRef> eingesetzt werden.

1535 **5.2.3.4 SIGDATAOBJURI**

1536 Anstelle der Variable `SIGDATAOBJURI` MUSS im XML-Signaturlayout der aus den
1537 Signaturparametern zusammengesetzte Wert `<ParamSigDataObjURI>` eingesetzt werden.

1538 **5.2.3.5 DIGESTVALUESIGNEDDATA**

1539 Anstelle der Variable `DIGESTVALUESIGNEDDATA` MUSS im XML-Signaturlayout der Hash-
1540 Wert der ersten XML-Signaturreferenz eingesetzt werden.

1541 **5.2.3.6 ETSIDATAREF**

1542 Anstelle der Variable `ETSIDATAREF` MUSS im XML-Signaturlayout der aus den
1543 Signaturparametern zusammengesetzte Wert `<ParamEtsiDataRef>` eingesetzt werden.

1544 **5.2.3.7 ETSIDATAOBJURI**

1545 Anstelle der Variable `ETSIDATAOBJURI` MUSS im XML-Signaturlayout der aus den
1546 Signaturparametern zusammengesetzte Wert `<ParamEtsiDataObjURI>` eingesetzt werden.

1547 **5.2.3.8 DIGESTVALUESIGNEDPROPERTIES**

1548 Anstelle der Variable `DIGESTVALUESIGNEDPROPERTIES` MUSS im XML-Signaturlayout der
1549 Hash-Wert der zweiten XML-Signaturreferenz eingesetzt werden.

1550 **5.2.3.9 SIGNATUREVALUE**

1551 Anstelle der Variable `SIGNATUREVALUE` MUSS im XML-Signaturlayout der Signaturwert
1552 eingesetzt werden.

1553 **5.2.3.10 X509CERTIFICATE**

1554 Anstelle der Variable `X509CERTIFICATE` MUSS im XML-Signaturlayout das Base64-kodierte
1555 Signaturzertifikat (X509-Zertifikat, kodiert gem. Abschnitt 4.4.4 in [4]) eingesetzt werden.

1556 **5.2.3.11 DSIGOBJECT**

1557 Wird eine Signatur-Methode angewandt, die eine enveloping XML-Signatur erwirkt, so MUSS
1558 das folgende Fragment anstelle der Variable `DSIGOBJECT` im XML-Signaturlayout eingesetzt
1559 werden:

```
1560 <dsig:Object Id="signed-data-object-SIGDATAOBJURI">  
1561 <dsig:Base64Content>BASE64CONTENT</dsig:Base64Content>  
1562 </dsig:Object>
```

1563 In diesem Fragment MUSS anstelle der Variable `SIGDATAOBJURI` der aus den
1564 Signaturparametern zusammengesetzte Wert `<ParamSigDataObjURI>` eingesetzt werden.

1565 In diesem Fragment MUSS anstelle der Variable `BASE64CONTENT` der Base64-kodierte, zu
1566 signierende Datenstrom (gem. angewandter Signaturmethode) eingesetzt werden.

1567 Wird eine Signatur-Methode angewandt, die eine detached XML-Signatur erwirkt, so ist kein
1568 XML-Signaturobjekt für die signierten Daten notwendig. Die Variable `DSIGOBJECT` im XML-
1569 Signaturlayout MUSS ersatzlos entfernt werden.

1570 **5.2.3.12 SIGNINGTIME**

1571 Anstelle der Variable `SIGNINGTIME` MUSS der Signaturzeitpunkt gem. [5] eingesetzt werden.

1572 **5.2.3.13 DIGESTVALUEX509CERTIFICATE**

1573 Anstelle der Variable `DIGESTVALUEX509CERTIFICATE` MUSS der Hash-Wert des
1574 Signaturzertifikates (lt. [5]) eingesetzt werden.

1575 **5.2.3.14 X509ISSUERNAME**

1576 Anstelle der Variable `X509ISSUERNAME` MUSS der Name des Ausstellers des
1577 Signaturzertifikates (lt. [5]) eingesetzt werden.

1578 **5.2.3.15 X509SERIALNUMBER**

1579 Anstelle der Variable `X509SERIALNUMBER` MUSS die Seriennummer des Signaturzertifikates
1580 (lt. [5]) eingesetzt werden.

1581 **5.3 Signaturparameter-Profil etsi-bka-1.0**

1582 **5.3.1 Charakteristik**

1583 *Parameter-Kennzeichnung:* etsi-bka-1.0

1584 *Signaturerstellungskomponente:* dieses Profil MUSS für die Generallizenz der
1585 Bürgerkartenumgebung (IT-Solution trustDesk-Basic), ab
1586 Version 2.7.5 angewendet werden.

1587 *Einschränkungen bzgl. Signatur-*
1588 *methoden:* keine, kann mit allen Signaturmethoden verwendet werden

1589 *Verwendbarkeit:* EMPFOHLEN

1590 **5.3.2 Signaturparameter**

1591 Für dieses Signaturparameter-Profil wurden Signaturparameter definiert. Diese MÜSSEN nach
1592 der vorangestellten Kennzeichnung des Profils in die Signatur-Repräsentation eingefügt
1593 werden.

1594 Für die Formulierung der Signaturparameter MUSS die Definition gem. Abschnitt 2.2 angewandt
1595 werden. Es gilt bei diesem Signaturparameter-Profil jedoch die Maßgabe, dass das Feld
1596 `<SIGDEV_PROF>` den Wert `etsi-bka-1.0` haben MUSS.

1597 Die Signaturparameter dieses Profils repräsentieren 5 Einzelwerte. Dabei MUSS als Parameter
1598 Teil 1 (Feld `PARAM_L1` der Signaturparameter, siehe 2.2) als konstanter Präfix für alle 5
1599 Einzelwerte herangezogen werden. Der Parameter Teil 2 (Feld `PARAM_L2` der
1600 Signaturparameter, siehe 2.2) enthält selbst 5 Einzelwerte, die jeweils mit einem Bindestrich
1601 separiert werden müssen. Es gilt für Parameter Teil 2 folgende Ausformung (in Ergänzung zur
1602 Definition aus Abschnitt 2.2):

```
1603 <PARAM_L2> ::= <WERT_1>"-"<WERT_2>"-"<WERT_3>"-"<WERT_4>"-"<WERT_5>  
1604 <WERT_1> ::= <CHAR>  
1605 <WERT_2> ::= <CHAR>  
1606 <WERT_3> ::= <CHAR>  
1607 <WERT_4> ::= <CHAR>  
1608 <WERT_5> ::= <CHAR>
```

1609 Daraus MÜSSEN folgende Einzelwerte gebildet werden:

```
1610 <ParamSigID> ::= <PARAM_L1>"-"<WERT_1>  
1611 <ParamSigDataRef> ::= "0-"<PARAM_L1>"-"<WERT_2>  
1612 <ParamSigDataObjURI> ::= "0-"<PARAM_L1>"-"<WERT_3>  
1613 <ParamEtsiDataRef> ::= "0-"<PARAM_L1>"-"<WERT_4>  
1614 <ParamEtsiDataObjURI> ::= "0-"<PARAM_L1>"-"<WERT_5>
```

1615 Dieser Werte werden mehrfach im Signaturlayout referenziert und verwendet. Wird eine
1616 Signatur-Methode angewandt, die eine detached XML-Signatur erzeugt, so wird der Wert
1617 <ParamSigDataObjURI> nicht benötigt. Die Signaturparameter lassen die Bildung dieses
1618 Wertes jedoch dennoch zu.

1619 Die Verwendung dieser Einzelwerte wird in Abschnitt 5.2.3 festgelegt.

1620 Nachfolgend ein Beispiel zur Bildung der Einzelwerte auf Basis übergebener Signaturparameter
1621 (spezifisch für das vorliegende Signaturparameter-Profil):

1622 Signaturparameter lt. Signaturrepräsentation:

1623 etsi-bka-1.0@1200412799-27800484@23524-22018-0-24095-30271

1624 Daraus ergeben sich:

1625 <ParamSigID> = 1200412799-27800484-23524
1626 <ParamSigDataRef> = 0-1200412799-27800484-22018
1627 <ParamSigDataObjURI> = 0-1200412799-27800484-0
1628 <ParamEtsiDataRef> = 0-1200412799-27800484-24095
1629 <ParamEtsiDataObjURI> = 0-1200412799-27800484-30271

1630 5.3.3 Signaturlayout

1631 Die Signaturen einer Bürgerkartenumgebung enthalten zeitabhängige Varianzen. Diese sind vor
1632 allem eine Reihe von XML-Attributen (ID-Attribute) die zur Referenzierung von XML-
1633 Elementen/-Knoten herangezogen werden. Diese variablen Attribute MÜSSEN in Form der
1634 Signaturparameter innerhalb der Signatur-Repräsentation verwaltet und bei der Rekonstruktion
1635 der Signatur entsprechend berücksichtigt werden.

1636 Eine XML-Signatur, die diesem Parameter-Profil entspricht, MUSS dem folgenden Layout
1637 entsprechen:

```
1638 <dsig:Signature xmlns:dsig="http://www.w3.org/2000/09/xmldsig#" Id="signature-SIGID">  
1639   <dsig:SignedInfo>  
1640     <dsig:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-  
1641     20010315"/>  
1642     <dsig:SignatureMethod Algorithm="ALGORITHM" />  
1643     <dsig:Reference Id="signed-data-reference-SIGDATAREF" URI="REFERENCE">  
1644       TRANSFORMS  
1645       <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>  
1646       <dsig:DigestValue>DIGESTVALUESIGNEDDATA</dsig:DigestValue>  
1647     </dsig:Reference>  
1648     <dsig:Reference Id="etsi-data-reference-ETSIDATAREF"  
1649     Type="http://uri.etsi.org/01903/v1.1.1#SignedProperties"  
1650     URI="#xmlns(etsi=http://uri.etsi.org/01903/v1.1.1%23)%20xpointer(id('etsi-data-object-  
1651     ETSIDATAOBJURI')/child::etsi:QualifyingProperties/child::etsi:SignedProperties)">  
1652       <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>  
1653       <dsig:DigestValue>DIGESTVALUESIGNEDPROPERTIES</dsig:DigestValue>  
1654     </dsig:Reference>  
1655   </dsig:SignedInfo>  
1656   <dsig:SignatureValue>SIGNATUREVALUE</dsig:SignatureValue>  
1657   <dsig:KeyInfo>  
1658     <dsig:X509Data>  
1659       <dsig:X509Certificate>X509CERTIFICATE</dsig:X509Certificate>  
1660     </dsig:X509Data>  
1661   </dsig:KeyInfo>  
1662   DSIGOBJECT  
1663   <dsig:Object Id="etsi-data-object-ETSIDATAOBJURI">  
1664     <etsi:QualifyingProperties xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"  
1665     xmlns:etsi="http://uri.etsi.org/01903/v1.1.1#" Target="#signature-SIGID">  
1666       <etsi:SignedProperties xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"  
1667       xmlns:etsi="http://uri.etsi.org/01903/v1.1.1#">  
1668         <etsi:SignedSignatureProperties>  
1669           <etsi:SigningTime>SIGNINGTIME</etsi:SigningTime>
```

```
1670 <etsi:SigningCertificate>
1671 <etsi:Cert>
1672 <etsi:CertDigest>
1673 <etsi:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
1674 <etsi:DigestValue>DIGESTVALUEX509CERTIFICATE</etsi:DigestValue>
1675 </etsi:CertDigest>
1676 <etsi:IssuerSerial>
1677 <dsig:X509IssuerName>X509ISSUERNAME</dsig:X509IssuerName>
1678 <dsig:X509SerialNumber>X509SERIALNUMBER</dsig:X509SerialNumber>
1679 </etsi:IssuerSerial>
1680 </etsi:Cert>
1681 </etsi:SigningCertificate>
1682 <etsi:SignaturePolicyIdentifier>
1683 <etsi:SignaturePolicyImplied/>
1684 </etsi:SignaturePolicyIdentifier>
1685 </etsi:SignedSignatureProperties>
1686 <etsi:SignedDataObjectProperties>
1687 <etsi:DataObjectFormat ObjectReference="#signed-data-reference-SIGDATAREF">
1688 <etsi:MimeType>MIMETYPE</etsi:MimeType>
1689 </etsi:DataObjectFormat>
1690 </etsi:SignedDataObjectProperties>
1691 </etsi:SignedProperties>
1692 </etsi:QualifyingProperties>
1693 </dsig:Object>
1694 </dsig:Signature>
```

1695 Die vom Layout vorgesehenen Variabilitäten (Variablen) werden in Großbuchstaben und
1696 umrandet ausgewiesen. Die nachfolgenden Abschnitte definieren diese im Detail.

1697 Dieses Profil legt durch das damit festgelegte XML-Signaturlayout die folgenden wesentlichen
1698 XML-Signatur-Eigenschaften implizit fest:

- 1699 1. Kanonisierungsmethode: <http://www.w3.org/TR/2001/REC-xml-c14n-20010315>
- 1700 2. Digest-Methode: <http://www.w3.org/2000/09/xmldsig#sha1>
- 1701 3. Qualifying Properties: nach ETSI (<http://uri.etsi.org/01903/v1.1.1#>)

1702 5.3.3.1 SIGID

1703 Anstelle der Variable **SIGID** MUSS im XML-Signaturlayout der aus den Signaturparametern
1704 zusammengesetzte Wert <ParamSigID> eingesetzt werden.

1705 5.3.3.2 ALGORITHM

1706 Legt den Signaturalgorithmus fest. Es MUSS einer der folgenden Signaturalgorithmen
1707 verwendet werden, in Abhängigkeit der Eigenschaften des Signatur-Schlüssels:

- 1708 • für ECDSA-Schlüssel: <http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha1>
- 1709 • für RSA-Schlüssel: <http://www.w3.org/2000/09/xmldsig#rsa-sha1>

1710 Der zum Signaturschlüssel passende Identifikator (URI) MUSS im XML-Signaturlayout anstelle
1711 der Variable **ALGORITHM** eingesetzt werden.

1712 5.3.3.3 SIGDATAREF

1713 Anstelle der Variable **SIGDATAREF** MUSS im XML-Signaturlayout der aus den
1714 Signaturparametern zusammengesetzte Wert <ParamSigDataRef> eingesetzt werden.

1715 5.3.3.4 SIGDATAOBJURI

1716 Anstelle der Variable **SIGDATAOBJURI** MUSS im XML-Signaturlayout der aus den
1717 Signaturparametern zusammengesetzte Wert <ParamSigDataObjURI> eingesetzt werden.

1718 Dieser Wert wird mehrfach im Signaturlayout referenziert und verwendet. Wird eine Signatur-
1719 Methode angewandt, die eine detached XML-Signatur erzeugt, so wird dieser Wert nicht
1720 benötigt. Die Signaturparameter lassen die Bildung dieses Wertes jedoch dennoch zu.

1721 5.3.3.5 REFERENCE

1722 Wird eine Signatur-Methode angewandt, die eine enveloping XML-Signatur erwirkt, so MUSS
1723 das folgende Fragment anstelle der Variable `REFERENCE` im XML-Signaturlayout eingesetzt
1724 werden:

```
1725 #signed-data-object-SIGDATAOBJURI
```

1726 In diesem Fragment MUSS anstelle der Variable `SIGDATAOBJURI` der aus den
1727 Signaturparametern zusammengesetzte Wert `<ParamSigDataObjURI>` eingesetzt werden.

1728 Beispiel: `#signed-data-object-0-1155648477-25748375-22389`

1729 Wird eine Signatur-Methode angewandt, die eine detached XML-Signatur erzeugt, so MUSS
1730 das folgende Fragment anstelle der Variable `REFERENCE` im XML-Signaturlayout eingesetzt
1731 werden:

```
1732 urn:Document
```

1733 5.3.3.6 TRANSFORMS

1734 Wird eine Signatur-Methode angewandt, die eine enveloping XML-Signatur erzeugt, so MUSS
1735 das folgende Fragment anstelle der Variable `TRANSFORMS` im XML-Signaturlayout eingesetzt
1736 werden:

```
1737 <dsig:Transforms>  
1738   <dsig:Transform Algorithm="http://www.w3.org/2002/06/xmldsig-filter2">  
1739     <xpf:XPath xmlns:xpf="http://www.w3.org/2002/06/xmldsig-filter2"  
1740       Filter="intersect">id('signed-data-object-SIGDATAOBJURI')/node()</xpf:XPath>  
1741     </dsig:Transform>  
1742     <dsig:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#base64"/>  
1743   </dsig:Transforms>
```

1744 In diesem Fragment MUSS anstelle der Variable `SIGDATAOBJURI` der aus den
1745 Signaturparametern zusammengesetzte Wert `<ParamSigDataObjURI>` eingesetzt werden.

1746 Wird eine Signatur-Methode angewandt, die eine detached XML-Signatur erwirkt, so ist kein
1747 Transformationspfad einzufügen. In diesem Fall DARF ein Transformationspfad NICHT im XML-
1748 Signaturlayout eingesetzt werden. Die Variable `TRANSFORMS` im XML-Signaturlayout MUSS
1749 ersatzlos entfernt werden.

1750 5.3.3.7 DIGESTVALUESIGNEDDATA

1751 Anstelle der Variable `DIGESTVALUESIGNEDDATA` MUSS im XML-Signaturlayout der Hash-
1752 Wert der ersten XML-Signaturreferenz eingesetzt werden.

1753 5.3.3.8 ETSIDATAREF

1754 Anstelle der Variable `ETSIDATAREF` MUSS im XML-Signaturlayout der aus den
1755 Signaturparametern zusammengesetzte Wert `<ParamEtsiDataRef>` eingesetzt werden.

1756 5.3.3.9 ETSIDATAOBJURI

1757 Anstelle der Variable `ETSIDATAOBJURI` MUSS im XML-Signaturlayout der aus den
1758 Signaturparametern zusammengesetzte Wert `<ParamEtsiDataObjURI>` eingesetzt werden.

1759 **5.3.3.10 DIGESTVALUESIGNEDPROPERTIES**

1760 Anstelle der Variable `DIGESTVALUESIGNEDPROPERTIES` MUSS im XML-Signaturlayout der
1761 Hash-Wert der zweiten XML-Signaturreferenz eingesetzt werden.

1762 **5.3.3.11 SIGNATUREVALUE**

1763 Anstelle der Variable `SIGNATUREVALUE` MUSS im XML-Signaturlayout der Signaturwert
1764 eingesetzt werden.

1765 **5.3.3.12 X509CERTIFICATE**

1766 Anstelle der Variable `X509CERTIFICATE` MUSS im XML-Signaturlayout das Base64-kodierte
1767 Signaturzertifikat (X509-Zertifikat, kodiert gem. Abschnitt 4.4.4 in [4]) eingesetzt werden.

1768 **5.3.3.13 DSIGOBJECT**

1769 Wird eine Signatur-Methode angewandt, die eine enveloping XML-Signatur erwirkt, so MUSS
1770 das folgende Fragment anstelle der Variable `DSIGOBJECT` im XML-Signaturlayout eingesetzt
1771 werden:

```
1772 <dsig:Object Id="signed-data-object-SIGDATAOBJURI">  
1773 <dsig:Base64Content>BASE64CONTENT</dsig:Base64Content>  
1774 </dsig:Object>
```

1775 In diesem Fragment MUSS anstelle der Variable `SIGDATAOBJURI` der aus den
1776 Signaturparametern zusammengesetzte Wert `<ParamSigDataObjURI>` eingesetzt werden.

1777 In diesem Fragment MUSS anstelle der Variable `BASE64CONTENT` der Base64-kodierte, zu
1778 signierende Datenstrom (gem. angewandter Signaturmethode) eingesetzt werden.

1779 Wird eine Signatur-Methode angewandt, die eine detached XML-Signatur erwirkt, so ist kein
1780 XML-Signaturobjekt für die signierten Daten notwendig. Die Variable `DSIGOBJECT` im XML-
1781 Signaturlayout MUSS ersatzlos entfernt werden.

1782 **5.3.3.14 SIGNINGTIME**

1783 Anstelle der Variable `SIGNINGTIME` MUSS der Signaturzeitpunkt gem. [5] eingesetzt werden.

1784 **5.3.3.15 DIGESTVALUEX509CERTIFICATE**

1785 Anstelle der Variable `DIGESTVALUEX509CERTIFICATE` MUSS der Hash-Wert des
1786 Signaturzertifikates (lt. [5]) eingesetzt werden.

1787 **5.3.3.16 X509ISSUENAME**

1788 Anstelle der Variable `X509ISSUENAME` MUSS der Name des Ausstellers des
1789 Signaturzertifikates (lt. [5]) eingesetzt werden.

1790 **5.3.3.17 X509SERIALNUMBER**

1791 Anstelle der Variable `X509SERIALNUMBER` MUSS die Seriennummer des Signaturzertifikates
1792 (lt. [5]) eingesetzt werden.

1793 **5.3.3.18 MIMETYPE**

1794 Anstelle der Variable `MIMETYPE` MUSS der MIME-Type der zu signierenden Daten eingesetzt
1795 werden. Der MIME-Type ist von der angewandten Signaturmethode abhängig und MUSS im
1796 Zuge der Definition der Signaturmethode festgelegt werden.

1797 **5.4 Signaturparameter-Profil *etsi-moc-1.0***

1798 **5.4.1 Charakteristik**

1799	<i>Parameter-Kennzeichnung:</i>	etsi-moc-1.0
1800	<i>Signaturerstellungskomponente:</i>	dieses Profil MUSS für die Open Source BKU "MOCCA"
1801		verwendet werden.
1802	<i>Einschränkungen bzgl. Signatur-</i>	urn:pdfsigfilter:bka.gv.at:text:v1.1.0,
1803	<i>methoden:</i>	urn:pdfsigfilter:bka.gv.at:binaer:v1.1.0
1804	<i>Verwendbarkeit:</i>	EMPFOHLEN

1805 **5.4.2 Signaturparameter**

1806 Für dieses Signaturparameter-Profil wurden Signaturparameter definiert. Diese MÜSSEN nach
1807 der vorangestellten Kennzeichnung des Profils in die Signatur-Repräsentation eingefügt
1808 werden.

1809 Für die Formulierung der Signaturparameter MUSS die Definition gem. Abschnitt 2.2 angewandt
1810 werden. Es gilt bei diesem Signaturparameter-Profil jedoch die Maßgabe, dass das Feld
1811 <SIGDEV_PROF> den Wert etsi-moc-1.0 haben MUSS.

1812 Der Signaturparameter PARAM_L1 (siehe Abschnitt 2.2) stellt einen jeweils innerhalb einer
1813 Signatur konstanten Wert dar, der für die Bildung der 7 unten beschriebenen Einzelwerte
1814 herangezogen werden MUSS. Der Parameter PARAM_L2 wird nicht verwendet und entfällt
1815 zusammen mit dem Delimiter-Zeichen "@".

1816 Aus dem Signaturparameter PARAM_L1 MÜSSEN folgende Einzelwerte gebildet werden:

1817	<ParamSigId>	::= "Signature-"<PARAM_L1>"-1"
1818	<ParamSignedInfoId>	::= "SignedInfo-"<PARAM_L1>"-1"
1819	<ParamSigDataRefId>	::= "Reference-"<PARAM_L1>"-1"
1820	<ParamEtsiDataRefId>	::= "Reference-"<PARAM_L1>"-2"
1821	<ParamSigValueId>	::= "SignatureValue-"<PARAM_L1>"-1"
1822	<ParamEtsiSignedPropertiesId>	::= "SignedProperties-"<PARAM_L1>"-1"
1823	<ParamEtsiDataObjId>	::= "Object-"<PARAM_L1>"-1"

1824 Diese Werte werden mehrfach im Signaturlayout referenziert und verwendet.

1825 Die Verwendung dieser Einzelwerte wird in Abschnitt 5.4.3 festgelegt.

1826 Nachfolgend ein Beispiel zur Bildung der Einzelwerte auf Basis übergebener Signaturparameter
1827 (spezifisch für das vorliegende Signaturparameter-Profil):

1828 Signaturparameter lt. Signaturrepräsentation:

1829 etsi-moc-1.0@b2e01c95

1830 Daraus ergeben sich:

1831	<ParamSigId>	= Signature-b2e01c95-1
1832	<ParamSignedInfoId>	= SignedInfo-b2e01c95-1
1833	<ParamSigDataRefId>	= Reference-b2e01c95-1
1834	<ParamEtsiDataRefId>	= Reference-b2e01c95-2
1835	<ParamSigValueId>	= SignatureValue-b2e01c95-1
1836	<ParamEtsiSignedPropertiesId>	= SignedProperties-b2e01c95-1
1837	<ParamEtsiDataObjId>	= Object-b2e01c95-1

1838 5.4.3 Signaturlayout

1839 Die Signaturen einer Bürgerkartenumgebung enthalten zeitabhängige Varianzen. Diese sind vor
1840 allem eine Reihe von XML-Attributen (ID-Attribute) die zur Referenzierung von XML-
1841 Elementen/-Knoten herangezogen werden. Diese variablen Attribute MÜSSEN in Form der
1842 Signaturparameter innerhalb der Signatur-Repräsentation verwaltet und bei der Rekonstruktion
1843 der Signatur entsprechend berücksichtigt werden.

1844 Eine XML-Signatur, die diesem Parameter-Profil entspricht, MUSS dem folgenden Layout
1845 entsprechen:

```
1846 <dsig:Signature xmlns:dsig="http://www.w3.org/2000/09/xmldsig#" Id="SIGID">
1847   <dsig:SignedInfo Id="SIGNEDINFOID">
1848     <dsig:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
1849     <dsig:SignatureMethod Algorithm="ALGORITHM" />
1850     <dsig:Reference Id="SIGDATAREFID" URI="urn:Document">
1851       <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
1852       <dsig:DigestValue>DIGESTVALUESIGNEDDATA</dsig:DigestValue>
1853     </dsig:Reference>
1854     <dsig:Reference Id="ETSIDATAREFID"
1855     Type="http://uri.etsi.org/01903/v1.1.1#SignedProperties"
1856     URI="#xmlns(xades=http://uri.etsi.org/01903/v1.1.1%23)%20xpointer(id('ETSIDATAOBJID'))/
1857     child::xades:QualifyingProperties/child::xades:SignedProperties)">
1858       <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
1859       <dsig:DigestValue>DIGESTVALUESIGNEDPROPERTIES</dsig:DigestValue>
1860     </dsig:Reference>
1861   </dsig:SignedInfo>
1862   <dsig:SignatureValue Id="SIGVALUEID">SIGNATUREVALUE</dsig:SignatureValue>
1863   <dsig:KeyInfo>
1864     <dsig:X509Data>
1865       <dsig:X509Certificate>X509CERTIFICATE</dsig:X509Certificate>
1866     </dsig:X509Data>
1867   </dsig:KeyInfo>
1868   <dsig:Object Id="ETSIDATAOBJID">
1869     <QualifyingProperties xmlns="http://uri.etsi.org/01903/v1.1.1#"
1870     xmlns:ns2="http://www.w3.org/2000/09/xmldsig#"
1871     <SignedProperties xmlns="http://uri.etsi.org/01903/v1.1.1#"
1872     xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
1873     xmlns:ns2="http://www.w3.org/2000/09/xmldsig#" Id="ETSISIGNEDPROPERTIESID">
1874       <SignedSignatureProperties>
1875         <SigningTime>SIGNINGTIME</SigningTime>
1876         <SigningCertificate>
1877           <Cert>
1878             <CertDigest>
1879               <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
1880               <DigestValue>DIGESTVALUEX509CERTIFICATE</DigestValue>
1881             </CertDigest>
1882             <IssuerSerial>
1883               <ns2:X509IssuerName>X509ISSUENAME</ns2:X509IssuerName>
1884               <ns2:X509SerialNumber>X509SERIALNUMBER</ns2:X509SerialNumber>
1885             </IssuerSerial>
1886           </Cert>
1887         </SigningCertificate>
1888         <SignaturePolicyIdentifier>
1889           <SignaturePolicyImplied />
1890         </SignaturePolicyIdentifier>
1891       </SignedSignatureProperties>
1892       <SignedDataObjectProperties>
1893         <DataObjectFormat ObjectReference="#SIGDATAREFID">
1894           <MimeType>MIMETYPE</MimeType>
1895         </DataObjectFormat>
1896       </SignedDataObjectProperties>
1897     </SignedProperties>
1898   </QualifyingProperties>
1899 </dsig:Object>
1900 </dsig:Signature>
```

1901 Die vom Layout vorgesehenen Variabilitäten (Variablen) werden in Großbuchstaben und
1902 umrandet ausgewiesen. Die nachfolgenden Abschnitte definieren diese im Detail.

1903 Dieses Profil legt durch das damit festgelegte XML-Signaturlayout die folgenden wesentlichen
1904 XML-Signatur-Eigenschaften implizit fest:

- 1905 1. Kanonisierungsmethode: <http://www.w3.org/2001/10/xml-exc-c14n#>
- 1906 2. Digest-Methode: <http://www.w3.org/2000/09/xmlsig#sha1>
- 1907 3. Qualifying Properties: nach ETSI (<http://uri.etsi.org/01903/v1.1.1#>)

1908 **5.4.3.1 SIGID**

1909 Anstelle der Variable **SIGID** MUSS im XML-Signaturlayout der aus dem Signaturparameter
1910 zusammengesetzte Wert `<ParamSigId>` eingesetzt werden.

1911 **5.4.3.2 SIGNEDINFOID**

1912 Anstelle der Variable **SIGNEDINFOID** MUSS im XML-Signaturlayout der aus dem
1913 Signaturparameter zusammengesetzte Wert `<ParamSignedInfoId>` eingesetzt werden.

1914 **5.4.3.3 ALGORITHM**

1915 Legt den Signaturalgorithmus fest. Es MUSS einer der folgenden Signaturalgorithmen
1916 verwendet werden, in Abhängigkeit der Eigenschaften des Signatur-Schlüssels:

- 1917 • für ECDSA-Schlüssel: <http://www.w3.org/2001/04/xmlsig-more#ecdsa-sha1>
- 1918 • für RSA-Schlüssel: <http://www.w3.org/2000/09/xmlsig#rsa-sha1>

1919 Der zum Signaturschlüssel passende Identifikator (URI) MUSS im XML-Signaturlayout anstelle
1920 der Variable **ALGORITHM** eingesetzt werden.

1921 **5.4.3.4 SIGDATAREFID**

1922 Anstelle der Variable **SIGDATAREFID** MUSS im XML-Signaturlayout der aus den
1923 Signaturparametern zusammengesetzte Wert `<ParamSigDataRefId>` eingesetzt werden.

1924 **5.4.3.5 DIGESTVALUESIGNEDDATA**

1925 Anstelle der Variable **DIGESTVALUESIGNEDDATA** MUSS im XML-Signaturlayout der Base64-
1926 kodierte Hash-Wert der ersten XML-Signaturreferenz eingesetzt werden.

1927 **5.4.3.6 ETSIDATAREFID**

1928 Anstelle der Variable **ETSIDATAREFID** MUSS im XML-Signaturlayout der aus den
1929 Signaturparametern zusammengesetzte Wert `<ParamEtsiDataRefId>` eingesetzt werden.

1930 **5.4.3.7 ETSIDATAOBJID**

1931 Anstelle der Variable **ETSIDATAOBJID** MUSS im XML-Signaturlayout der aus den
1932 Signaturparametern zusammengesetzte Wert `<ParamEtsiDataObjId>` eingesetzt werden.

1933 **5.4.3.8 DIGESTVALUESIGNEDPROPERTIES**

1934 Anstelle der Variable **DIGESTVALUESIGNEDPROPERTIES** MUSS im XML-Signaturlayout der
1935 Base64-kodierte Hash-Wert der zweiten XML-Signaturreferenz eingesetzt werden.

1936 **5.4.3.9 SIGVALUEID**

1937 Anstelle der Variable **SIGVALUEID** MUSS im XML-Signaturlayout der aus den
1938 Signaturparametern zusammengesetzte Wert <ParamSigValueId> eingesetzt werden.

1939 **5.4.3.10 SIGNATUREVALUE**

1940 Anstelle der Variable **SIGNATUREVALUE** MUSS im XML-Signaturlayout der Base64-kodierte
1941 Signaturwert eingesetzt werden.

1942 **5.4.3.11 X509CERTIFICATE**

1943 Anstelle der Variable **X509CERTIFICATE** MUSS im XML-Signaturlayout das Base64-kodierte
1944 Signaturzertifikat (X509-Zertifikat, kodiert gem. Abschnitt 4.4.4 in [4]) eingesetzt werden.

1945 **5.4.3.12 ETSISIGNEDPROPERTIESID**

1946 Anstelle der Variable **ETSI SIGNED PROPERTIES ID** MUSS im XML-Signaturlayout der aus den
1947 Signaturparametern zusammengesetzte Wert <ParamEtsiSignedPropertiesId>
1948 eingesetzt werden.

1949 **5.4.3.13 SIGNINGTIME**

1950 Anstelle der Variable **SIGNINGTIME** MUSS der Signaturzeitpunkt gem. [5] eingesetzt werden.

1951 **5.4.3.14 DIGESTVALUEX509CERTIFICATE**

1952 Anstelle der Variable **DIGESTVALUEX509CERTIFICATE** MUSS der Base64-kodierte Hash-
1953 Wert des Signaturzertifikates (lt. [5]) eingesetzt werden.

1954 **5.4.3.15 X509ISSUERNAME**

1955 Anstelle der Variable **X509ISSUERNAME** MUSS der Name des Ausstellers des
1956 Signaturzertifikates (lt. [5]) eingesetzt werden.

1957 **5.4.3.16 X509SERIALNUMBER**

1958 Anstelle der Variable **X509SERIALNUMBER** MUSS die Seriennummer des Signaturzertifikates
1959 (lt. [5]) eingesetzt werden.

1960 **5.4.3.17 MIMETYPE**

1961 Anstelle der Variable **MIMETYPE** MUSS der MIME-Type der zu signierenden Daten eingesetzt
1962 werden. Der MIME-Type ist von der angewandten Signaturmethode abhängig und MUSS im
1963 Zuge der Definition der Signaturmethode festgelegt werden.

1964

1965

1966 **Referenzen**

- 1967 [1] ISO-19005-1:2005: Document management - Electronic document file format for long-
1968 term preservation - Part 1: Use of PDF 1.4 (PDF/A-1). ISO Standard 19005-1:2005.
- 1969 [2] Adobe Corporation: PDF Reference, third edition - Adobe portable document format
1970 version 1.4, ISBN 0-201-75839-3, December 2001, Addison-Wesley.
- 1971 [3] Adobe Corporation: PDF Reference, fifth edition - Adobe Portable Document Format
1972 version 1.6.
- 1973 [4] Eastlake, Donald, Reagle, Joseph und Solo, David: XML-Signature Syntax and
1974 Processing. W3C Recommendation, Februar 2002. Abgerufen aus dem World Wide
1975 Web am 14.05.2004 unter <http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/>.
- 1976 [5] ETSI XML Advanced Electronic Signatures (XAdES): ETSI TS 101 903 V1.2.2 (2004-
1977 04), Technical Specification.
- 1978 [6] Hollosi, Karlinger, et.al.: Die österreichische Bürgerkarte , Spezifikationsframework
1979 Version 1.2, März 2005. Abgerufen aus dem World Wide Web am 18.01.2008 unter
1980 <http://www.buergerkarte.at/konzept/securitylayer/spezifikation/aktuell/>.
- 1981 [7] Stabsstelle IKT-Strategie des Bundes: Spezifikation Module für Online Anwendungen –
1982 SP und SS, Version 1.3.0, vom 28. August 2005. Abgerufen aus dem World Wide Web
1983 am 10.01.2008 unter <http://egovlabs.gv.at/docman/view.php/6/20/MOA-SPSS-1.3.pdf>.
- 1984 [8] Rössler: Layout Amtssignatur – Spezifikation. Version 1.0.0 vom 25.06.2007.
- 1985 [9] RFC 2234: Augmented BNF for Syntax Specifications: ABNF