

PDF Signatur/Amtssignatur Spezifikation

Version 2.2

Inhalt

PDF Signatur/Amtssignatur	1
Spezifikation	1
Dokument-Historie	3
Schlüsselwörter	4
Begriffserklärungen	5
1 Einleitung	7
2 Charakteristika von PDF-Amtssignaturen	8
2.1 Methode	8
2.2 Parameter	9
2.3 Anforderungen an PDF Dokumente	11
3 Repräsentation einer PDF-Amtssignatur	12
4 Definierte Signaturmethoden	13
4.1 Textuelle Signatur, Version 1.0.0	14
4.2 Textuelle Signatur, Version 1.1.0	18
4.3 Textuelle Signatur, Version 1.2.0	21
4.4 Binäre Signatur, Version 1.0.0	22
4.5 Binäre Signatur, Version 1.1.0	32
5 Definierte Signaturparameter	41
5.1 Default Signaturparameter-Profil	41
5.2 Default Signaturparameter-Profil für BKU	45
5.3 Signaturparameter-Profil etsi-bka-1.0	50
5.4 Signaturparameter-Profil etsi-moc-1.0	55
5.5 Signaturparameter-Profil etsi-bka-atrust-1.0	59
6 PDF Signature Field	62
7 BAIK Archiv Spezifikation	63
7.1 Darstellung des Signaturblocks	63
7.2 Erweiterungen zur PDF-AS Spezifikation	65
Referenzen	66

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."

Datum	Version	Autor / Organisation	Änderungen
26.10.2010	2.2	David Ferbas (e-nnovation)	Hinzufügen des Parameters SIGDEV_SPEC. Erweiterung der Signaturparameter-Profile um variable Algorithmen-Suiten und Hashmethoden (SIGDEV_SPEC) Neues Signaturparameter-Profil „etsi-bka-atrust-1.0“ Neue Signaturmethode: Textuelle Signatur, Version 1.2.0
30.09.2010	2.2	Thomas Knall (EGIZ)	Screenshot für Beurkundungssignatur ausgetauscht

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 Signaturerstellungszusammenhangs 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:v1.2.0

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]) angewandt werden:

```

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

```

<PARAMETER_ID>	Der Signaturparameter.
<SIGDEV_PROF>	Kennzeichnung der Signaturerstellungskomponente sowie eines optionalen Signaturparameter-Profiles.
<SIGDEV_SPEC>	Spezifikation der verwendeten Signatur-Suite und Hashalgorithmen. Details siehe Abschnitt 2.2.1
<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.

152 **Beispiele:**

153 etsi-bka-1.0@1155648477-25748375@3902-22389-0-16078-16384
 154 etsi-bka-1.1:ecdsa-sha1@1204789497-2215265@3074-32627-0-18335-
 155 29801
 156 etsi-moc-1.1:ecdsa-sha1:ripemd160@207c44ff
 157 moass-1.0@1234ABC@
 158 etsi-moc-1.0@b62b3b59
 159 foo@@1234abcd

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

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

162 **2.2.1 Signaturspezifikation SIGDEV_SPEC**

163 Der Parameter SIGDEV_SPEC KANN zur näheren Beschreibung der verwendeten
 164 Signaturerstellungseinheit verwendet werden sofern die verwendete Signatursuite bzw. die
 165 verwendeten Hashalgorithmen vom jeweiligen Standardverfahren des Signaturparameters
 166 abweichen. Über diesen Parameter ist die Angabe einer Signatursuite bzw. von
 167 Hashalgorithmen möglich. Es werden eine Signatursuite und drei Hashalgorithmen (Data-
 168 Digest, Property-Digest, Zertifikat-Digest) definiert:

169 *Signatur-Suite : Data-Hashverfahren : Properties-Hashverfahren : Certificate-Hashverfahren*

170 Der erste Parameter, steht für die Signatur-Suite und MUSS vorhanden sein. Die nächsten drei
 171 stehen für die jeweils verwendeten Hashalgorithmen (für die erste, bzw. für die zweite Referenz
 172 sowie für den Digest des Zertifikats) und SOLLEN vorzugsweise entfallen, falls sämtliche
 173 verwendete Hashalgorithmen jenem der Signatur-Suite entsprechen bzw. falls nachfolgende
 174 Hashalgorithmen keine Änderungen zu den vorhergehenden Algorithmen darstellen.

175 Folgende Tabelle dient als Beispiel für diese abgekürzte Schreibweise.

176

Lange Schreibweise	Äquivalente kurze Schreibweise
ecdsa-sha1:sha1:sha1:sha1	ecdsa-sha1
ecdsa-sha256:sha1:sha1:sha1	ecdsa-sha256:sha1
ecdsa-sha256:sha256:sha1:sha1	ecdsa-sha256:sha256:sha1

177

178 Um die Signatur-Parameter kurz zu halten werden die Kurzbezeichnungen (z.B. sha1) bei der
179 Rekonstruktion auf entsprechende URIs (z.B. <http://www.w3.org/2000/09/xmlsig#sha1>)
180 zurückgeführt.

181 Es MÜSSEN zumindest folgende Verfahren unterstützt werden

182

183 **Signatur-Methoden**

184 <http://www.w3.org/2000/09/xmlsig#dsa-sha1>
185 <http://www.w3.org/2000/09/xmlsig#hmac-sha1>
186 <http://www.w3.org/2000/09/xmlsig#rsa-sha1>
187 <http://www.w3.org/2001/04/xmlsig-more#ecdsa-sha1>
188 <http://www.w3.org/2001/04/xmlsig-more#ecdsa-sha224>
189 <http://www.w3.org/2001/04/xmlsig-more#ecdsa-sha256>
190 <http://www.w3.org/2001/04/xmlsig-more#ecdsa-sha384>
191 <http://www.w3.org/2001/04/xmlsig-more#ecdsa-sha512>
192 <http://www.w3.org/2001/04/xmlsig-more#hmac-md5>
193 <http://www.w3.org/2001/04/xmlsig-more#hmac-ripemd160>
194 <http://www.w3.org/2001/04/xmlsig-more#hmac-sha224>
195 <http://www.w3.org/2001/04/xmlsig-more#hmac-sha256>
196 <http://www.w3.org/2001/04/xmlsig-more#hmac-sha384>
197 <http://www.w3.org/2001/04/xmlsig-more#hmac-sha512>
198 <http://www.w3.org/2001/04/xmlsig-more#rsa-md5>
199 <http://www.w3.org/2001/04/xmlsig-more#rsa-ripemd160>
200 <http://www.w3.org/2001/04/xmlsig-more#rsa-sha256>
201 <http://www.w3.org/2001/04/xmlsig-more#rsa-sha384>
202 <http://www.w3.org/2001/04/xmlsig-more#rsa-sha512>
203 <http://www.w3.org/2007/05/xmlsig-more#ecdsa-ripemd160>

204

205 **Digest-Algorithmen**

206 <http://www.w3.org/2000/09/xmlsig#sha1>
207 <http://www.w3.org/2001/04/xmlsig-more#md5>
208 <http://www.w3.org/2001/04/xmlsig-more#sha224>
209 <http://www.w3.org/2001/04/xmlsig-more#sha384>
210 <http://www.w3.org/2001/04/xmlenc#ripemd160>
211 <http://www.w3.org/2001/04/xmlenc#sha256>
212 <http://www.w3.org/2001/04/xmlenc#sha512>

213 **2.3 Anforderungen an PDF Dokumente**

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

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

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

227 3 Repräsentation einer PDF-Amtssignatur

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

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

Signaturwert	XX	
	Unterzeichner	XX
	Datum/Zeit-UTC	XX
	Aussteller-Zertifikat	XX
	Serien-Nr.	XX
	Methode	XX
	Parameter	XX
Prüfhinweis	XX	

239 **Abbildung 1: Muster einer visuellen Ausprägung der Amtssignatur**

240 Das Layout bzw. die Anordnung der einzelnen Felder sowie die Bezeichnung der Felder KANN
241 frei gewählt werden. Semantisch MÜSSEN die folgenden Vorgaben für eine visuellen
242 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.

#	Feld-Bezeichnung (Signaturattribut)	M/K/S	Beschreibung
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.

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

247 4 Definierte Signaturmethoden

248 Die vorliegende Spezifikation definiert eine Reihe von Signaturmethoden, die wie folgt in
249 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	DEPRECATED	EMPFOHLEN	NICHT EMPFOHLEN
urn:pdfsigfilter:bka.gv.at:text:v1.2.0	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

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

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

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

259 **4.1 Textuelle Signatur, Version 1.0.0**

260 **4.1.1 Charakteristik**

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

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

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

266 *Art der Signatur:* XML-Signatur, Enveloping Signature

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

268 *Anwendbarkeit:* NICHT EMPFOHLEN
269 deprecated, wurde ersetzt durch
270 urn:pdfsigfilter:bka.gv.at:text:v1.1.0

271 **4.1.2 Aufbereitung der zu signierenden Daten**

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

274 Der Input-Datenstrom MUSS wie folgt behandelt werden:

- 275 1. Der Input-Datenstrom (das PDF-Dokument) wird geöffnet.
- 276 2. Es wird der Text des gegebenen Originaldokuments extrahiert. Dabei MÜSSEN folgende
277 Vorgaben beachtet werden:
 - 278 a. der extrahierte Text MUSS eine Zeichenfolge sein, die den auf dem PDF-Dokument
279 dargestellten Text entspricht.
 - 280 b. die Zeichenfolge MUSS der Leserichtung folgend von links oben nach rechts unten
281 aufgelöst werden.
 - 282 c. Besonderheiten der PDF-Repräsentation, wie etwa die Darstellung fett gedruckter
283 Text-Teile durch Überlappung leicht versetzter Einzelzeichen, MÜSSEN ignoriert
284 und auf den eigentliche Textinhalt reduziert werden.
- 285 3. Auf den extrahierten Text MÜSSEN die folgenden Normalisierungsmaßnahmen in der hier
286 festgelegten Reihenfolge angewendet werden:
 - 287 a. Alle NULL-Zeichen (\u0000) werden entfernt.
 - 288 b. Alle Tabulatoren (\u0009) und Seitenumbrüche (\u000C) werden durch einzelne
289 Leerzeichen (\u0020) ersetzt.
 - 290 c. Alle No-Break Spaces (\u00A0) werden durch Leerzeichen (\u0020) ersetzt.

- 291 d. Alle Vorkommnisse von Zeilenumbrüchen (Newlines) – systemabhängig, zum
292 Beispiel bei Windows die Kombination der Zeichen Zeilenumbruch (\u000D) und
293 Zeilenvorschub (\u000A) bzw. bei MacOS nur das Zeichen Zeilenvorschub (\u000A)
294 – werden durch ein Zeichen Zeilenvorschub (\u000A) ersetzt.
- 295 e. Mehrfache Zeilenumbrüche, das heißt zwei oder mehrere, werden auf zwei
296 Zeilenumbrüche (zwei Zeichen \u000A) reduziert.
- 297 f. Alle mehrfachen Leerzeichen (\u0020) werden durch ein einfaches Leerzeichen
298 (\u0020) ersetzt.
- 299 g. Leerzeichen (\u0020) am Zeilenanfang oder am Zeilenende werden entfernt.
- 300 h. Leerzeilen, das sind Zeilen ohne jeglichen Inhalt bzw. die nur mehr ein Leerzeichen
301 enthalten, am Anfang bzw. am Ende des gesamten Textes (Dokuments) werden
302 entfernt.
- 303 i. Alle Arten von Apostrophen (Zeichen wie \u0060, \u00B4, \u2018, \u2019, \u201A,
304 \u201B) werden durch das Zeichen Apostroph (\u0027) ersetzt.
- 305 j. Alle Arten von Anführungsstriche (Zeichen wie \u201C, \u201D, \u201E, \u201F)
306 werden durch das Zeichen Anführungszeichen (\u0022) ersetzt.
- 307 k. Alle Arten von Bindestriche (Zeichen wie \u00AD, \u2013, \u2014) werden durch das
308 Zeichen Bindestrich (\u002D) ersetzt.

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

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

316 4.1.3 XML-Signaturformat

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

320 Der Transformationspfad MUSS die folgenden Transformationen in dieser Reihenfolge
321 enthalten:

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

324 Die zu erstellende XML-Signatur ist eine Enveloping Signature gem. [4], welche in Form eines
325 Datenobjekts die signierten Daten eingebettet enthält. Diese MÜSSEN Base-64 kodiert als
326 `dsig:Object` Element in die XML-Signatur eingebettet werden (näheres dazu siehe [4] und
327 [6]).

328 Die erstellte XML-Signatur folgt den Vorgaben des österreichischen E-Governments bzw. den
329 Vorgaben für XML-Signaturen aus der Spezifikation der österreichischen Bürgerkarte (siehe
330 [6]).

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

```
333 <dsig:Signature Id="signature-1161003152-26578093-24674"  
334 xmlns:dsig="http://www.w3.org/2000/09/xmldsig#">  
335   <dsig:SignedInfo>  
336     <dsig:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-  
337     20010315"/>  
338     <dsig:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#ecdsa-  
339     sha1"/>  
340     <dsig:Reference Id="signed-data-reference-0-1161003152-26578093-5873" URI="#signed-  
341     data-object-0-1161003152-26578093-8480">  
342       <dsig:Transforms>  
343         <dsig:Transform Algorithm="http://www.w3.org/2002/06/xmldsig-filter2">  
344           <xpf:XPath Filter="intersect" xmlns:xpf="http://www.w3.org/2002/06/xmldsig-  
345     filter2">id(&apos;signed-data-object-0-1161003152-26578093-  
346     8480&apos;)/node()/</xpf:XPath>  
347         </dsig:Transform>  
348         <dsig:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#base64"/>  
349       </dsig:Transforms>  
350       <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>  
351       <dsig:DigestValue>HASH-WERT DER 1. REFERENZ</dsig:DigestValue>  
352     </dsig:Reference>  
353     <dsig:Reference Id="etsi-data-reference-0-1161003152-26578093-26221"  
354     Type="http://uri.etsi.org/01903/v1.1.1#SignedProperties" URI="#etsi-data-object-0-  
355     1161003152-26578093-25255">  
356       <dsig:Transforms>  
357         <dsig:Transform Algorithm="http://www.w3.org/2002/06/xmldsig-filter2">  
358           <xpf:XPath Filter="intersect" xmlns:xpf="http://www.w3.org/2002/06/xmldsig-  
359     filter2">id(&apos;etsi-data-object-0-1161003152-26578093-  
360     25255&apos;)/node()/</xpf:XPath>  
361         </dsig:Transform>  
362       </dsig:Transforms>  
363       <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>  
364       <dsig:DigestValue>HASH-WERT DER 2. REFERENZ</dsig:DigestValue>  
365     </dsig:Reference>  
366   </dsig:SignedInfo>  
367   <dsig:SignatureValue>SIGNATURWERT</dsig:SignatureValue>  
368   <dsig:KeyInfo>  
369     <dsig:X509Data>  
370       <dsig:X509Certificate>ZERTIFIKAT</dsig:X509Certificate>  
371     </dsig:X509Data>  
372   </dsig:KeyInfo>  
373   <dsig:Object Id="signed-data-object-0-1161003152-26578093-8480">  
374     <sl:Base64Content>SIGNIERTE DATEN (BASE64)</sl:Base64Content>  
375   </dsig:Object>  
376   <dsig:Object Id="etsi-data-object-0-1161003152-26578093-25255">  
377     <etsi:QualifyingProperties Target="#signature-1161003152-26578093-24674"  
378     xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"&br/>379     xmlns:etsi="http://uri.etsi.org/01903/v1.1.1#">  
380       <etsi:SignedProperties>  
381         <etsi:SignedSignatureProperties>  
382           <etsi:SigningTime>SIGNATURZEITPUNKT</etsi:SigningTime>  
383           <etsi:SigningCertificate>  
384             <etsi:Cert>  
385               <etsi:CertDigest>  
386                 <etsi:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>  
387                 <etsi:DigestValue>HASHWERT DES SIGNATURZERTIFIKATES</etsi:DigestValue>  
388               </etsi:CertDigest>  
389               <etsi:IssuerSerial>  
390                 <dsig:X509IssuerName>AUSSTELLER DES ZERTIFIKATS</dsig:X509IssuerName>  
391                 <dsig:X509SerialNumber>SERIENNUMMER DES ZERTIFIKATS</dsig:X509SerialNumber>  
392               </etsi:IssuerSerial>  
393             </etsi:Cert>  
394           </etsi:SigningCertificate>  
395           <etsi:SignaturePolicyIdentifier>  
396             <etsi:SignaturePolicyImplied/>  
397           </etsi:SignaturePolicyIdentifier>  
398         </etsi:SignedSignatureProperties>
```

```
399     <etsi:SignedDataObjectProperties>  
400         <etsi:DataObjectFormat ObjectReference="#signed-data-reference-0-1161003152-  
401 26578093-5873">  
402             <etsi:MimeType>text/plain</etsi:MimeType>  
403         </etsi:DataObjectFormat>  
404     </etsi:SignedDataObjectProperties>  
405 </etsi:SignedProperties>  
406 </etsi:QualifyingProperties>  
407 </dsig:Object>  
408 </dsig:Signature>
```

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

413 **4.1.4 Einbettung der Signatur in das PDF-Dokument**

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

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

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

424 **4.1.5 Anwendungshinweis zur Verifikation**

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

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

432 Die Vorgehensweise der Verifikation im Überblick:

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

448 **4.2 Textuelle Signatur, Version 1.1.0**

449 **4.2.1 Charakteristik**

450	<i>Methoden-Kennzeichnung:</i>	urn:pdfsigfilter:bka.gv.at:text:v1.1.0
451	<i>Input-Datenstrom:</i>	das zu signierende PDF-Dokument
452		(binärer Datenstrom, application/pdf)
453	<i>Signierte Datenstrom:</i>	der aus dem PDF-Dokument extrahierte Text
454		(binärer Datenstrom, text/plain)
455	<i>Art der Signatur:</i>	XML-Signatur, Detached Signature
456	<i>Zulässige Signaturparameter:</i>	keine Einschränkung
457	<i>Anwendbarkeit:</i>	NICHT EMPFOHLEN
458		deprecated, wurde ersetzt durch
459		urn:pdfsigfilter:bka.gv.at:text:v1.2.0

460 **4.2.2 Aufbereitung der zu signierenden Daten**

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

463 Der Input-Datenstrom MUSS wie folgt behandelt werden:

- 464 1. Der Input-Datenstrom (das PDF-Dokument) wird geöffnet.
- 465 2. Es wird der Text des gegebenen Originaldokuments extrahiert. Dabei MÜSSEN folgende
466 Vorgaben beachtet werden:
 - 467 a. der extrahierte Text MUSS eine Zeichenfolge sein, die den auf dem PDF-Dokument
468 dargestellten Text entspricht.
 - 469 b. die Zeichenfolge MUSS der Leserichtung folgend von links oben nach rechts unten
470 aufgelöst werden.
 - 471 c. Besonderheiten der PDF-Repräsentation, wie etwa die Darstellung fett gedruckter
472 Text-Teile durch Überlappung leicht versetzter Einzelzeichen, MÜSSEN ignoriert
473 und auf den eigentliche Textinhalt reduziert werden.
- 474 3. Auf den extrahierten Text MÜSSEN die folgenden Normalisierungsmaßnahmen in der hier
475 festgelegten Reihenfolge angewendet werden:
 - 476 a. Alle NULL-Zeichen (\u0000) werden entfernt.
 - 477 b. Alle Tabulatoren (\u0009) und Seitenumbrüche (\u000C) werden durch einzelne
478 Leerzeichen (\u0020) ersetzt.
 - 479 c. Alle No-Break Spaces (\u00A0) werden durch Leerzeichen (\u0020) ersetzt.
 - 480 d. Alle Vorkommnisse von Zeilenumbrüchen (Newlines) – systemabhängig, zum
481 Beispiel bei Windows die Kombination der Zeichen Zeilenumbruch (\u000D) und
482 Zeilenvorschub (\u000A) bzw. bei MacOS nur das Zeichen Zeilenvorschub (\u000A)
483 – werden durch ein Zeichen Zeilenvorschub (\u000A) ersetzt.
 - 484 e. Mehrfache Zeilenumbrüche, das heißt zwei oder mehrere, werden auf zwei
485 Zeilenumbrüche (zwei Zeichen \u000A) reduziert.
 - 486 f. Alle mehrfachen Leerzeichen (\u0020) werden durch ein einfaches Leerzeichen
487 (\u0020) ersetzt.
 - 488 g. Leerzeichen (\u0020) am Zeilenanfang oder am Zeilenende werden entfernt.

- 489 h. Leerzeilen, das sind Zeilen ohne jeglichen Inhalt bzw. die nur mehr ein Leerzeichen
490 enthalten, am Anfang bzw. am Ende des gesamten Textes (Dokuments) werden
491 entfernt.
- 492 i. Alle Arten von Apostrophen (Zeichen wie \u0060, \u00B4, \u2018, \u2019, \u201A,
493 \u201B) werden durch das Zeichen Apostroph (\u0027) ersetzt.
- 494 j. Alle Arten von Anführungsstriche (Zeichen wie \u201C, \u201D, \u201E, \u201F)
495 werden durch das Zeichen Anführungszeichen (\u0022) ersetzt.
- 496 k. Alle Arten von Bindestriche (Zeichen wie \u00AD, \u2013, \u2014) werden durch das
497 Zeichen Bindestrich (\u002D) ersetzt.

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

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

505 4.2.3 XML-Signaturformat

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

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

510 Die erstellte XML-Signatur folgt den Vorgaben des österreichischen E-Governments bzw. den
511 Vorgaben für XML-Signaturen aus der Spezifikation der österreichischen Bürgerkarte (siehe
512 [6]).

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

```
515 <dsig:Signature xmlns:dsig="http://www.w3.org/2000/09/xmldsig#" Id="signature-  
516 1201005938-3018328-13744">  
517   <dsig:SignedInfo>  
518     <dsig:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-  
519 20010315"/>  
520     <dsig:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#ecdsa-  
521 sha1"/>  
522     <dsig:Reference Id="signed-data-reference-0-1201005938-3018328-20478"  
523 URI="urn:Document">  
524       <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>  
525       <dsig:DigestValue>HASH-WERT DER 1. REFERENZ</dsig:DigestValue>  
526     </dsig:Reference>  
527     <dsig:Reference Id="etsi-data-reference-0-1201005938-3018328-327"  
528 Type="http://uri.etsi.org/01903/v1.1.1#SignedProperties"  
529 URI="#xmlns(etsi=http://uri.etsi.org/01903/v1.1.1%23)%20xpointer(id('etsi-data-object-  
530 0-1201005938-3018328-  
531 18413'))/child::etsi:QualifyingProperties/child::etsi:SignedProperties)">  
532       <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>  
533       <dsig:DigestValue>HASH-WERT DER 2. REFERENZ</dsig:DigestValue>  
534     </dsig:Reference>  
535   </dsig:SignedInfo>  
536   <dsig:SignatureValue>SIGNATURWERT</dsig:SignatureValue>  
537   <dsig:KeyInfo>  
538     <dsig:X509Data>  
539       <dsig:X509Certificate>ZERTIFIKAT</dsig:X509Certificate>  
540     </dsig:X509Data>  
541   </dsig:KeyInfo>
```

```
542 <dsig:Object Id="etsi-data-object-0-1201005938-3018328-18413">
543   <etsi:QualifyingProperties xmlns:etsi="http://uri.etsi.org/01903/v1.1.1#"
544   Target="#signature-1201005938-3018328-13744">
545     <etsi:SignedProperties>
546       <etsi:SignedSignatureProperties>
547         <etsi:SigningTime>SIGNATURZEITPUNKT</etsi:SigningTime>
548         <etsi:SigningCertificate>
549           <etsi:Cert>
550             <etsi:CertDigest>
551               <etsi:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
552               <etsi:DigestValue>HASH-WERT DES ZERTIFIKATS</etsi:DigestValue>
553             </etsi:CertDigest>
554             <etsi:IssuerSerial>
555               <dsig:X509IssuerName>AUSSTELLER DES ZERTIFIKATS</dsig:X509IssuerName>
556               <dsig:X509SerialNumber>SERIENNUMMER DES ZERTIFIKATS</dsig:X509SerialNumber>
557             </etsi:IssuerSerial>
558           </etsi:Cert>
559         </etsi:SigningCertificate>
560         <etsi:SignaturePolicyIdentifier>
561           <etsi:SignaturePolicyImplied/>
562         </etsi:SignaturePolicyIdentifier>
563       </etsi:SignedSignatureProperties>
564       <etsi:SignedDataObjectProperties>
565         <etsi:DataObjectFormat ObjectReference="#signed-data-reference-0-1201005938-
566 3018328-20478">
567           <etsi:MimeType>text/plain</etsi:MimeType>
568         </etsi:DataObjectFormat>
569       </etsi:SignedDataObjectProperties>
570     </etsi:SignedProperties>
571   </etsi:QualifyingProperties>
572 </dsig:Object>
573 </dsig:Signature>
```

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

578 4.2.4 Einbettung der Signatur in das PDF-Dokument

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

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

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

589 4.2.5 Anwendungshinweis zur Verifikation

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

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

597 Die Vorgehensweise der Verifikation im Überblick:

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

613 **4.3 Textuelle Signatur, Version 1.2.0**

614 **4.3.1 Charakteristik**

615 <i>Methoden-Kennzeichnung:</i>	urn:pdfsigfilter:bka.gv.at:text:v1.2.0
616 <i>Input-Datenstrom:</i>	das zu signierende PDF-Dokument 617 (binärer Datenstrom, application/pdf)
618 <i>Signierte Datenstrom:</i>	der aus dem PDF-Dokument extrahierte Text 619 (binärer Datenstrom, text/plain)
620 <i>Art der Signatur:</i>	XML-Signatur, Detached Signature
621 <i>Zulässige Signaturparameter:</i>	keine Einschränkung
622 <i>Anwendbarkeit:</i>	EMPFOHLEN

623 **4.3.2 Unterschied zur Version 1.1.0**

624 Der Unterschied zur Vorgängerversion 1.1.0 besteht in der Behandlung von Glyphen, die nicht
625 über ein Font-Mapping d.h. nicht über einen Font-Descriptor in eine Unicode-Repräsentation
626 überführt werden können.

627 Für solche Glyphen sieht die Methode 1.2.0 vor, deren Byte-Darstellung für die eine
628 Substitution nach UTF-8 heranzuziehen.

629 **4.4 Binäre Signatur, Version 1.0.0**

630 **4.4.1 Charakteristik**

631	<i>Methoden-Kennzeichnung:</i>	urn:pdfsigfilter:bka.gv.at:binaer:v1.0.0
632	<i>Input-Datenstrom:</i>	das zu signierende PDF-Dokument
633		(binärer Datenstrom, application/pdf)
634	<i>Signierter Datenstrom:</i>	das aufbereitete PDF-Dokument
635		(binärer Datenstrom, application/pdf)
636	<i>Art der Signatur:</i>	XML-Signatur, Enveloping Signature
637	<i>Zulässige Signaturparameter:</i>	Default (MOA), Default (BKU), etsi-bka-1.0
638	<i>Anwendbarkeit:</i>	NICHT EMPFOHLEN
639		deprecated, wurde ersetzt durch
640		urn:pdfsigfilter:bka.gv.at:binaer:v1.1.0

641 **4.4.2 Aufbereitung der zu signierenden Daten**

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

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

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

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

- 656 1. Dem PDF-Dokument MUSS die Signatur-Repräsentation (Signaturblock) bereits vor der
657 Signaturerstellung eingebettet werden. Dazu ist gemäß den Vorgaben aus Abschnitt 4.4.4
658 der Signaturblock erstellt und in das Dokument eingebracht werden. Anstelle der zu
659 diesem Zeitpunkt noch unbekanntes Werte (Signaturattribute wie Signaturwert,
660 Signaturzeitpunkt, Angaben zum Signaturzertifikat, etc.) MÜSSEN durch semantisch
661 wertfreie Füllzeichen ersetzt werden. Als Füllzeichen MUSS das NULL-Byte (numerisch 0)
662 verwendet werden (dies ist das Default Füllzeichen für die in der Signatur-Repräsentation
663 vorgesehenen Wertebereiche zur Fassung der Signaturattribute; siehe dazu auch die
664 Vorgaben aus Abschnitt 4.4.4.1.6). Nach erfolgter Signatur werden gemäß den Vorgaben
665 aus Abschnitt 4.4.4 die Signaturattribute in den durch Füllzeichen vorbereiteten
666 Wertebereiche der Signatur-Repräsentation eingefüllt.

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

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

677 4.4.3 XML-Signaturformat

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

681 Der Transformationspfad MUSS die folgenden Transformationen in dieser Reihenfolge
682 enthalten:

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

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

690 Die erstellte XML-Signatur folgt den Vorgaben des österreichischen E-Governments bzw. den
691 Vorgaben für XML-Signaturen aus der Spezifikation der österreichischen Bürgerkarte (siehe
692 [6]).

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

```
695 <dsig:Signature Id="signature-1161003152-26578093-24674"  
696 xmlns:dsig="http://www.w3.org/2000/09/xmldsig#">  
697   <dsig:SignedInfo>  
698     <dsig:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-  
699     20010315"/>  
700     <dsig:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#ecdsa-  
701     sha1"/>  
702     <dsig:Reference Id="signed-data-reference-0-1161003152-26578093-5873" URI="#signed-  
703     data-object-0-1161003152-26578093-8480">  
704       <dsig:Transforms>  
705         <dsig:Transform Algorithm="http://www.w3.org/2002/06/xmldsig-filter2">  
706           <xpf:XPath Filter="intersect" xmlns:xpf="http://www.w3.org/2002/06/xmldsig-  
707     filter2">id(&apos;signed-data-object-0-1161003152-26578093-  
708     8480&apos;)/node()/</xpf:XPath>  
709         </dsig:Transform>  
710         <dsig:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#base64"/>  
711       </dsig:Transforms>  
712       <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>  
713       <dsig:DigestValue>HASH-WERT DER 1. REFERENZ</dsig:DigestValue>  
714     </dsig:Reference>  
715     <dsig:Reference Id="etsi-data-reference-0-1161003152-26578093-26221"  
716     Type="http://uri.etsi.org/01903/v1.1.1#SignedProperties" URI="#etsi-data-object-0-  
717     1161003152-26578093-25255">  
718       <dsig:Transforms>  
719         <dsig:Transform Algorithm="http://www.w3.org/2002/06/xmldsig-filter2">  
720           <xpf:XPath Filter="intersect" xmlns:xpf="http://www.w3.org/2002/06/xmldsig-  
721     filter2">id(&apos;etsi-data-object-0-1161003152-26578093-  
722     25255&apos;)/node()/</xpf:XPath>  
723         </dsig:Transform>  
724       </dsig:Transforms>
```

```
725     <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
726     <dsig:DigestValue>HASH-WERT DER 2. REFERENZ</dsig:DigestValue>
727   </dsig:Reference>
728 </dsig:SignedInfo>
729 <dsig:SignatureValue>SIGNATURWERT</dsig:SignatureValue>
730 <dsig:KeyInfo>
731   <dsig:X509Data>
732     <dsig:X509Certificate>ZERTIFIKAT</dsig:X509Certificate>
733   </dsig:X509Data>
734 </dsig:KeyInfo>
735 <dsig:Object Id="signed-data-object-0-1161003152-26578093-8480">
736   <sl:Base64Content>SIGNIERTE DATEN (BASE64)</sl:Base64Content>
737 </dsig:Object>
738 <dsig:Object Id="etsi-data-object-0-1161003152-26578093-25255">
739   <etsi:QualifyingProperties Target="#signature-1161003152-26578093-24674"
740 xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
741 xmlns:etsi="http://uri.etsi.org/01903/v1.1.1#">
742     <etsi:SignedProperties>
743       <etsi:SignedSignatureProperties>
744         <etsi:SigningTime>SIGNATURZEITPUNKT</etsi:SigningTime>
745         <etsi:SigningCertificate>
746           <etsi:Cert>
747             <etsi:CertDigest>
748               <etsi:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
749               <etsi:DigestValue>HASH-WERT DES SIGNATURZERTIFIKATS</etsi:DigestValue>
750             </etsi:CertDigest>
751             <etsi:IssuerSerial>
752               <dsig:X509IssuerName>AUSSTELLER DES ZERTIFIKATS</dsig:X509IssuerName>
753               <dsig:X509SerialNumber>SERIENNUMMER DES ZERTIFIKATS</dsig:X509SerialNumber>
754             </etsi:IssuerSerial>
755           </etsi:Cert>
756         </etsi:SigningCertificate>
757         <etsi:SignaturePolicyIdentifier>
758           <etsi:SignaturePolicyImplied/>
759         </etsi:SignaturePolicyIdentifier>
760       </etsi:SignedSignatureProperties>
761       <etsi:SignedDataObjectProperties>
762         <etsi:DataObjectFormat ObjectReference="#signed-data-reference-0-1161003152-
763 26578093-5873">
764           <etsi:MimeType>text/plain</etsi:MimeType>
765         </etsi:DataObjectFormat>
766       </etsi:SignedDataObjectProperties>
767     </etsi:SignedProperties>
768   </etsi:QualifyingProperties>
769 </dsig:Object>
770 </dsig:Signature>
```

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

775 4.4.4 Einbettung der Signatur in das PDF-Dokument

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

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

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

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

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

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

799 4.4.4.1 EGIZ-Dictionary

800 Die Keys des EGIZ Dictionaries MÜSSEN, falls verwendet, in folgender Reihenfolge vorhanden
801 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.

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 4.4.4.1.2.</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 4.4.4.1.3.</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 4.4.4.1.4.</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 4.4.4.1.5.</p>

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

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

806 **4.4.4.1.1 /ID**

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

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

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

813 Zusammengefügt MÜSSEN die durch diese Byte-Ranges spezifizierten Strings wieder den
814 Signaturmethoden-String ergeben. Das Character Encoding des Kennzeichnungsstrings MUSS
815 8 bit WinAnsiEncoding sein. Zu beachten sind auch PDF String Escape Sequences (siehe
816 Kapitel „Strings in den Löchern“).

817 **4.4.4.1.2 /ByteRange**

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

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

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

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

834 **Beispiel:**

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

837 [0 100 110 90]

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

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

844 **Beispiel:**
 845 [...]

846 1 0 0 1 137.91 207 Tm

847 /F1 12 Tf

848 (123456789a) Tj

849 ET

850 BT

851 1 0 0 1 217.11 225 Tm

852 /F2 12 Tf

853 (123bcdefgh) Tj

854 1 0 0 1 217.11 213 Tm

855 [...]

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

861 **4.4.4.1.3 /replaces**

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

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

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

870 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).

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

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

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

885 **Beispiel:**

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

887 **4.4.4.1.4 /encodings**

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

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

895

896 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.

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

899 Wird eine Zeichenkette (Signaturattribut) aus Platzgründen auf mehrere Aussparungen
900 aufgeteilt (siehe Abschnitt 4.4.4.1.3), so MUSS für alle Fragmente dieser Zeichenkette das für
901 das erste Fragment festgelegte Encoding (festgelegt durch das zur entsprechenden
902 Aussparung korrespondierende Element des /encoding-Feldes) angenommen werden. Sind für
903 die anderen Fragmente einer Zeichenkette davon abweichende Encodings festgelegt
904 (festgelegt durch das zur entsprechenden Aussparung korrespondierende Element des
905 /encoding-Feldes), so MÜSSEN diese ignoriert werden.

906 Das nachfolgende Beispiel zeigt ein exemplarisches /encoding-Feld, im Einklang mit den zuvor,
907 im Beispiel des Abschnitts 4.4.4.1.3, definierten Aussparungen.

908 **Beispiel:**

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

910 4.4.4.1.5 /Cert

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

917 4.4.4.1.6 Zeichenketten (Strings) als Wert

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

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

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

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

937 1. „\“ werden durch „\\“ ersetzt.

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

939 3. „)“ werden durch „\)“ ersetzt.

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

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

947 **4.4.5 Anwendungshinweis zur Verifikation**

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

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

955 Die Vorgehensweise der Verifikation im Überblick:

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

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

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

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

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

975 **4.5 Binäre Signatur, Version 1.1.0**

976 **4.5.1 Charakteristik**

977 <i>Methoden-Kennzeichnung:</i>	urn:pdfsigfilter:bka.gv.at:binaer:v1.1.0
978 <i>Input-Datenstrom:</i>	das zu signierende PDF-Dokument 979 (binärer Datenstrom, application/pdf)
980 <i>Signierte Datenstrom:</i>	das aufbereitet PDF-Dokument 981 (binärer Datenstrom, application/pdf)
982 <i>Art der Signatur:</i>	XML-Signatur, Detached Signature
983 <i>Zulässige Signaturparameter:</i>	keine Einschränkung
984 <i>Anwendbarkeit:</i>	EMPFOHLEN

985 **4.5.2 Aufbereitung der zu signierenden Daten**

986 Der Aufbereitungsprozess ist aus Sicht des Signaturerstellungsprozesses definiert. Bei der
987 Verifikation ist analog vorzugehen (siehe auch Hinweis in Abschnitt 4.5.5).

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

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

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

- 1000 1. Dem PDF-Dokument MUSS die Signatur-Repräsentation (Signaturblock) bereits vor der
1001 Signaturerstellung eingebettet werden. Dazu ist gemäß den Vorgaben aus Abschnitt 4.4.4
1002 der Signaturblock erstellt und in das Dokument eingebracht werden. Anstelle der zu
1003 diesem Zeitpunkt noch unbekanntem Werte (Signaturattribute wie Signaturwert,
1004 Signaturzeitpunkt, Angaben zum Signaturzertifikat, etc.) MÜSSEN durch semantisch
1005 wertfreie Füllzeichen ersetzt werden. Als Füllzeichen MUSS das NULL-Byte (numerisch 0)
1006 verwendet werden (dies ist das Default Füllzeichen für die in der Signatur-Repräsentation
1007 vorgesehenen Wertebereiche zur Fassung der Signaturattribute; siehe dazu auch die
1008 Vorgaben aus Abschnitt 4.4.4.1.6). Nach erfolgter Signatur werden gemäß den Vorgaben
1009 aus Abschnitt 4.4.4 die Signaturattribute in den durch Füllzeichen vorbereiteten
1010 Wertebereiche der Signatur-Repräsentation eingefüllt.

1011 Das so vorbereitete PDF-Dokument wird als binärer Datenstrom (Octet Stream) interpretiert und
1012 als Datenstrom für die Signaturerstellung herangezogen. Dieser Datenstrom wird signiert.

1013 Der MIME-Type des zu signierenden Datenstroms MUSS daher im Rahmen der XML-Signatur
1014 auf application/pdf gesetzt werden. Dementsprechend MUSS in den erstellten XML-
1015 Signaturen, sofern diese Angaben zu Eigenschaften des signierten Dokumentes beinhalten
1016 (z.B. durch das Element etsi:SignedDataObjectProperties/
1017 etsi:DataObjectFormat) enthalten, der MIME-Type mit application/pdf angegeben
1018 werden.

1019 4.5.3 XML-Signaturformat

1020 Die resultierende Signatur ist eine XML Signatur nach [4]. Die zu signierenden Daten MÜSSEN
1021 nach Aufbereitung ohne weitere Veränderung als zu signierenden Daten für die Bildung der
1022 XML-Signatur herangezogen werden.

1023 Die zu erstellende XML-Signatur MUSS eine Detached Signature sein (siehe [4]).

1024 Die erstellte XML-Signatur folgt den Vorgaben des österreichischen E-Governments bzw. den
1025 Vorgaben für XML-Signaturen aus der Spezifikation der österreichischen Bürgerkarte (siehe
1026 [6]).

1027 Beispiel einer XML-Signatur nach diesen Vorgaben (erstellt mit der Bürgerkartensoftware IT-
1028 Solution trustDesk basic):

```
1029 <dsig:Signature xmlns:dsig="http://www.w3.org/2000/09/xmldsig#" Id="signature-  
1030 1201006039-3118750-12731">  
1031   <dsig:SignedInfo>  
1032     <dsig:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-  
1033 20010315"/>  
1034     <dsig:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#ecdsa-  
1035 sha1"/>  
1036     <dsig:Reference Id="signed-data-reference-0-1201006039-3118750-13501"  
1037 URI="urn:Document">  
1038       <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>  
1039       <dsig:DigestValue>HASH-WERT 1. REFERENZ</dsig:DigestValue>  
1040     </dsig:Reference>  
1041     <dsig:Reference Id="etsi-data-reference-0-1201006039-3118750-21710"  
1042 Type="http://uri.etsi.org/01903/v1.1.1#SignedProperties"  
1043 URI="#xmlns(etsi=http://uri.etsi.org/01903/v1.1.1%23)%20xpointer(id('etsi-data-object-  
1044 0-1201006039-3118750-  
1045 21127'))/child::etsi:QualifyingProperties/child::etsi:SignedProperties)">  
1046       <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>  
1047       <dsig:DigestValue>HASH-WERT 2. REFERENZ</dsig:DigestValue>  
1048     </dsig:Reference>  
1049   </dsig:SignedInfo>  
1050   <dsig:SignatureValue>SIGNATURWERT</dsig:SignatureValue>  
1051   <dsig:KeyInfo>  
1052     <dsig:X509Data>  
1053       <dsig:X509Certificate>ZERTIFIKAT</dsig:X509Certificate>  
1054     </dsig:X509Data>  
1055   </dsig:KeyInfo>  
1056   <dsig:Object Id="etsi-data-object-0-1201006039-3118750-21127">  
1057     <etsi:QualifyingProperties xmlns:etsi="http://uri.etsi.org/01903/v1.1.1#"   
1058 Target="#signature-1201006039-3118750-12731">  
1059       <etsi:SignedProperties>  
1060         <etsi:SignedSignatureProperties>  
1061           <etsi:SigningTime>SIGNATURZEITPUNKT</etsi:SigningTime>  
1062           <etsi:SigningCertificate>  
1063             <etsi:Cert>  
1064               <etsi:CertDigest>  
1065                 <etsi:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>  
1066                 <etsi:DigestValue>HASH-WERT DES ZERTIFIKATS</etsi:DigestValue>  
1067               </etsi:CertDigest>
```

```
1068     <etsi:IssuerSerial>  
1069     <dsig:X509IssuerName>AUSSTELLER DES ZERTIFIKATS</dsig:X509IssuerName>  
1070     <dsig:X509SerialNumber>SERIENNUMMER DES ZERTIFIKATS</dsig:X509SerialNumber>  
1071     </etsi:IssuerSerial>  
1072     </etsi:Cert>  
1073     </etsi:SigningCertificate>  
1074     <etsi:SignaturePolicyIdentifier>  
1075     <etsi:SignaturePolicyImplied/>  
1076     </etsi:SignaturePolicyIdentifier>  
1077     </etsi:SignedSignatureProperties>  
1078     <etsi:SignedDataObjectProperties>  
1079     <etsi:DataObjectFormat ObjectReference="#signed-data-reference-0-1201006039-  
1080 3118750-13501">  
1081     <etsi:MimeType>application/pdf</etsi:MimeType>  
1082     </etsi:DataObjectFormat>  
1083     </etsi:SignedDataObjectProperties>  
1084     </etsi:SignedProperties>  
1085     </etsi:QualifyingProperties>  
1086     </dsig:Object>  
1087 </dsig:Signature>
```

1088 Dieses Beispiel enthält einige Besonderheiten der Signaturerstellungskomponente
1089 (Bürgerkartensoftware und Signaturerstellungseinheit), auf die in Verbindung mit
1090 Signaturparametern noch eingegangen wird. Aus Gründen der Übersichtlichkeit wurden
1091 variable Inhalte größtenteils durch verbale Umschreibungen ersetzt (eingerahmter Text).

1092 4.5.4 Einbettung der Signatur in das PDF-Dokument

1093 Die resultierende XML-Signatur MUSS letztlich in Form der in Abschnitt 3 definierten
1094 Repräsentation in das PDF-Dokument integriert werden.

1095 Die Einbettung der Signatur-Repräsentation (Signaturblock) im PDF-Dokument MUSS mit Hilfe
1096 eines Inkrementellen Update Blocks (Incremental Update Block, Abschnitt 3.4.5 in [2]) realisiert
1097 werden. Dieser Block MUSS folgende Struktur aufweisen:

- 1098 1. Dieser Incremental Update Block MUSS ein eigenes Dictionary, das EGIZ-Dictionary,
1099 enthalten. Dieses MUSS ein indirektes Objekt sein.
- 1100 2. Der gesamte sichtbare Signaturblock MUSS in ein XObject Form eingebettet sein. (siehe
1101 /SigXObject Key des EGIZ Dictionaries, Abschnitt 4.5.4.1)
- 1102 3. Das trailer-Dictionary des Incremental Update Blocks MUSS einen Key /EGIZSigDict
1103 enthalten. Wert dieses Keys MUSS eine indirekte Referenz auf das EGIZ-Dictionary sein.

1104 Der nachfolgende Abschnitt beschreibt das EGIZ-Dictionary im Detail.

1105 Als Spezifikum dieses Algorithmus MUSS die Signatur-Repräsentation bereits vor dem
1106 Signaturvorgang, im Zuge der Aufbereitung der zu signierenden Daten, in das PDF-Dokument
1107 eingebracht werden. Anstelle der zu diesem Zeitpunkt noch unbekanntenen Werte, wie bspw.
1108 Signaturattribute (das sind zum Beispiel Signaturwert, Signaturzeitpunkt, Angaben zum
1109 Signaturzertifikat, etc.), MÜSSEN die dafür vorgesehenen Wertebereiche mit semantisch
1110 wertfreien Füllzeichen aufgefüllt werden.

1111 Das mit diesem vorbereiteten aber leeren Signaturblock versehene PDF-Dokument wird in
1112 seiner binären Repräsentation elektronisch signiert. Nach dem Signaturprozess MÜSSEN die
1113 dabei ermittelten Werte (Signaturattribute) in die dafür vorgesehenen Wertebereiche der
1114 Signatur-Repräsentation (in den nachfolgenden Abschnitten auch als ‚Ausparung‘ bezeichnet)
1115 eingefüllt werden.

1116

1117 **4.5.4.1 EGIZ-Dictionary**

1118 Die Keys des EGIZ Dictionaries MÜSSEN, falls verwendet, in folgender Reihenfolge vorhanden
1119 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 4.5.4.1.1.
/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 4.5.4.1.2.
/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 4.5.4.1.3.
/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 4.5.4.1.4.

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 4.5.4.1.5.

1120 Sämtliche Werte, falls nicht anders spezifiziert, MÜSSEN direkte Objekte, also direkt im EGIZ-
1121 Dictionary eingebettet sein.

1122 Das EGIZ-Dictionary DARF noch weitere Elemente enthalten. Diese können dazu verwendet
1123 werden die Signatur mit zusätzlicher Information auszustatten.

1124 4.5.4.1.1 /ID

1125 Das /ID-Element beschreibt die Byte-Ranges (siehe Beschreibung /ByteRange) der
1126 Zeichenkette (String), die die angewendete Signaturmethode (Methode) identifiziert.

1127 Der so identifizierte String ist jener Teil des Content Streams der Signatur-Repräsentation,
1128 welcher den Text der Signaturmethode enthält. Dieser String unterliegt genauso den PDF
1129 Formatierungsregeln und wird daher im Allgemeinen in mehrere PDF-Strings gebrochen sein.

1130 Die /ID Byte-Ranges beschreiben den Inhalt dieser PDF-Strings des Content Streams.

1131 Zusammengefügt MÜSSEN die durch diese Byte-Ranges spezifizierten Strings wieder den
1132 Signaturmethoden-String ergeben. Das Character Encoding des Kennzeichnungsstrings MUSS
1133 8 bit WinAnsiEncoding sein. Zu beachten sind auch PDF String Escape Sequences (siehe
1134 Kapitel „Strings in den Löchern“).

1135 4.5.4.1.2 /ByteRange

1136 Statische Bereiche werden durch Byte-Ranges beschrieben. Byte-Ranges sind in Analogie zu
1137 Byte-Ranges in Adobe PDF-Signaturen definiert (siehe PDF Reference 1.6 [3], Kapitel 8.7):

- 1138 • Eine Byte-Range MUSS aus dem Zahlentupel Startoffset und Länge (in Bytes) bestehen.
- 1139 • Beide MÜSSEN positive Ganzzahlen sein, wobei der Startoffset auch 0 sein DARF.
- 1140 • Der Startoffset MUSS vom Anfang der PDF-Datei an gemessen werden.
- 1141 • Die Länge MUSS die Anzahl an Bytes ab dem Startoffset angeben, welche zur Byte-
1142 Range gehören.

1143 In einer binären Signatur MÜSSEN alle statischen Bereiche mittels Byte-Ranges identifiziert
1144 werden. Die variablen Bereiche zwischen den Byte-Ranges werden als Aussparungen
1145 bezeichnet, da die dadurch binär identifizierten Bereiche des PDF-Dokumentes nicht von der
1146 Signatur abgedeckt werden. In die Bereiche dieser Aussparungen MÜSSEN zum
1147 Signierzeitpunkt sogenannte Platzhalter-Zeichen eingefügt sein. Nach erfolgter Signatur
1148 MÜSSEN diese durch die resultierenden Werte der Signatur (zum Beispiel Signaturwert,
1149 Signaturzeitpunkt, etc.) ersetzt werden.

1150 Die Angaben von Byte-Ranges in den Elementen der binären Signatur MÜSSEN immer gemäß
1151 ihrer Startoffsets aufsteigend sortiert sein.

1152 **Beispiel:**

1153 Die Byte-Range (10, 5) beschreibt die Bytes an den Positionen 10, 11, 12, 13 und 14. Ein
1154 korrespondierendes /ByteRange-Array könnte folgendermaßen aussehen:

1155 [0 100 110 90]

1156 Dieses beschreibt zwei Byte-Ranges – von 0 bis 99 sowie von 110 bis 199 – und eine
1157 Aussparung – von 100 bis 109.

1158 Mit Hilfe dieser Byte-Range Angaben werden die signierten Bereiche des Dokuments auf Byte-
1159 Ebene identifiziert. Umgekehrt werden damit jene, nichtsignierten Aussparungen identifiziert, in
1160 die nach der Signatur die Signaturattribute eingebettet werden müssen. Das nachfolgende
1161 Beispiel soll dies illustrieren.

1162 **Beispiel:**

```
1163     [...]
1164     1 0 0 1 137.91 207 Tm
1165     /F1 12 Tf
1166     ([123456789a) Tj
1167     ET
1168     BT
1169     1 0 0 1 217.11 225 Tm
1170     /F2 12 Tf
1171     ([123bcdefgh) Tj
1172     1 0 0 1 217.11 213 Tm
1173     [...]
```

1174 Die nicht umrahmten Bereiche sind jene Bereiche, die über die Angabe von Byte-Ranges
1175 als die „signierten Bytes“ des Dokuments identifiziert werden würden. Die umrahmten
1176 Bereiche stellen hingegen Aussparungen dar, welche im signierten Dokument durch
1177 semantisch wertfreie Platzhalter ersetzt werden. Diese Struktur wird durch das
1178 /ByteRange-Element auf Byte-Ebene beschrieben.

1179 **4.5.4.1.3 /replaces**

1180 Das Feld stellt ein PDF Array von PDF-Names dar, welche angeben, welches Signaturattribut in
1181 den Aussparungen der binären Signatur – binär repräsentiert durch die inversen Byte-Ranges
1182 des /ByteRange-Elements – enthalten sind.

1183 Es MUSS so viele Elemente im /replaces Array geben wie es Aussparungen durch die
1184 getroffene Definition der Byte-Ranges gibt. Gibt es in einem Dokument n Byte-Ranges, so
1185 enthält das /replaces-Array (n-1) Einträge für die (n-1) Aussparungen.

1186 Jedes Element im /replaces Array MUSS ein gültiger PDF-Name sein, welcher den
1187 semantischen Wert (Typ) des Inhalts der Aussparung festlegt.

1188 Folgende PDF-Namen sind zur Beschreibung der semantischen Werte definiert:

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

PDF-Name	Semantischer Wert (Typ) / 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).
/alg	Optional. KANN für die nähere Spezifikation der verwendeten Signatur-Suite verwendet werden. Wir nur im Rahmen der BAIK Spezifikation (Abschnitt 7) interpretiert, sonst entsprechend /nil zu behandeln.

1189 Unbekannte Typen MÜSSEN im Zuge der Verifikation der Signatur als nicht
1190 Spezifikationskonform zurückgewiesen werden.

1191 Signaturattribute, bzw. die sie repräsentierenden Zeichenketten, DÜRFEN aus Platzgründen
1192 auf mehrere, aufeinanderfolgende Aussparungen aufgeteilt werden. Der Inhalt von derart
1193 aufeinander folgenden Aussparungen gleichen Typs stellt zusammengefasst den Gesamtwert
1194 des betreffenden Signaturattributs dar. Um diesen Gesamtwert zu bilden MÜSSEN die
1195 Fragmente konkateniert werden.

1196 Das folgende Beispiel zeigt die Beschreibung der semantischen Inhalte von Aussparungen,
1197 wobei die erste Aussparung ein nicht näher spezifizierte Datenblock ist (Typ /nil), die zweite
1198 und die dritte Aussparung Fragmente der Zeichenkette des Signaturdatums enthalten (Typ
1199 /dat), die Aussparung vier bis sechs Fragmente der Zeichenkette des Signaturwertes
1200 enthalten (Typ /val), die siebte und achte Aussparung wiederum nicht näher spezifizierte
1201 Datenblöcke enthalten, die letzte Aussparung die die Signaturparameter beschreibende
1202 Zeichenkette repräsentiert (Typ /sid).

1203 **Beispiel:**

1204 `[/nil /dat /dat /val /val /val /nil /nil /sid]`

1205 **4.5.4.1.4 /encodings**

1206 Die Festlegung der Typen der in Aussparungen gefassten Werte durch das /replaces-Feld
1207 werden durch die analoge Festlegung von Kodierungsarten (Encoding) ergänzt. Das
1208 /encodings-Feld legt daher fest, auf welche Art die in den jeweiligen Aussparungen gefassten
1209 Daten (vor allem Zeichenketten) kodiert worden sind.

1210 Es MUSS für jeden Wert, der in einer Aussparung nach der Signatur eingebettet und im
1211 /replaces-Feld entsprechend festgelegt wurde, spezifiziert werden, auf welche Art dieser Wert
1212 kodiert ist.

1213 Es DÜRFEN die folgenden Encodings verwendet werden:

Name	Bedeutung (Encoding)
/nil	Die Zeichenkette ist unkodiert und als Binärdaten zu interpretieren.
/win	Die Zeichenkette ist mittels PDF WinAnsiEncoding (8bit) codiert. (siehe PDF Reference 1.4 [2], Appendix D).

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

1214 Unbekannte Encoding-Arten MÜSSEN im Zuge der Verifikation der Signatur als nicht
1215 spezifikationskonform zurückgewiesen werden.

1216 Wird eine Zeichenkette (Signaturattribut) aus Platzgründen auf mehrere Aussparungen
1217 aufgeteilt (siehe Abschnitt 4.4.4.1.3), so MUSS für alle Fragmente dieser Zeichenkette das für
1218 das erste Fragment festgelegte Encoding (festgelegt durch das zur entsprechenden
1219 Aussparung korrespondierende Element des /encoding-Feldes) angenommen werden. Sind für
1220 die anderen Fragmente einer Zeichenkette davon abweichende Encodings festgelegt
1221 (festgelegt durch das zur entsprechenden Aussparung korrespondierende Element des
1222 /encoding-Feldes), so MÜSSEN diese ignoriert werden.

1223 Das nachfolgende Beispiel zeigt ein exemplarisches /encoding-Feld, im Einklang mit den zuvor,
1224 im Beispiel des Abschnitts 4.4.4.1.3, definierten Aussparungen.

1225 **Beispiel:**

1226 `[/nil /f16 /nil /nil /win /win /url]`

1227 **4.5.4.1.5 /Cert**

1228 Dieses Element wurde in Synergie mit dem /Cert Element von Adobe PDF Signaturen
1229 definiert (siehe PDF Reference 1.6 [3], Tabelle 8.98). Es gilt die Einschränkung, dass, falls
1230 vorhanden, der Wert immer ein Feld von direkten Literal-String-Objekten sein MUSS. Jeder
1231 String MUSS ein Base64-codiertes Zertifikat im PEM Format enthalten. Das erste Zertifikat
1232 MUSS das Signaturzertifikat sein. Die restlichen Zertifikate stellen die Zertifikatskette zu einem
1233 vertrauenswürdigen Wurzelzertifikat dar und sind OPTIONAL.

1234 **4.5.4.1.6 Zeichenketten (Strings) als Wert**

1235 Layoutbedingt wird eine längere Zeichenkette der Signatur-Repräsentation auf mehrere
1236 Aussparungen aufgeteilt. Umgekehrt muss in der visuellen Darstellung der Signatur-
1237 Repräsentation (Signaturblock) im PDF selbst genügend Raum geschaffen werden, um den
1238 gesamte Zeichenkette aufnehmen zu können. Dies bedeutet, dass der theoretisch mögliche
1239 Platz in Aussparungen nicht voll ausgefüllt werden kann, da diese Zeichenkette visuell, im
1240 gewählten Layout des Signaturblocks keinen Platz mehr findet (Darstellung würde bspw. Über
1241 den Papierrand reichen).

1242 Um die in einer Aussparung gefassten Zeichen/Werte vom ungenutzten Raum der
1243 Aussparungen unterscheiden zu können, MÜSSEN alle nicht genutzten Bereiche der
1244 Aussparung mit dem NULL-Byte (numerisch 0) befüllt werden. Das NULL-Byte DARF NICHT in
1245 einzusetzenden Werten vorkommen.

1246 Vor dem Einsetzen der Werte (kodierte Zeichenketten) in die Bereiche der Aussparungen
1247 MÜSSEN, unabhängig der verwendeten und im Feld /encoding festgelegten Kodierung,
1248 allfällige im einzusetzenden Wert (Zeichenkette) PDF-Steuerzeichen maskiert werden.

1249 Die zu maskierenden PDF-Steuerzeichen sind der Backslash („\“) sowie die linke und rechte
1250 Rundklammer („(“ und „)“). Es MUSS jedes im einzusetzenden Wert (Zeichenkette)
1251 vorkommende PDF-Steuerzeichen durch einen Backslash eingeleitet (escaped) werden. Es
1252 MUSS daher vor dem Einsetzen der Zeichenkette/Werte folgende Ersetzung vorgenommen
1253 werden:

- 1254 1. „\“ werden durch „\\“ ersetzt.
- 1255 2. „(“ werden durch „\ („,“ ersetzt.
- 1256 3. „)“ werden durch „\)“ ersetzt.

1257 Bei der Rekonstruktion der Werte im Zuge der Verifikation MUSS diese Transformation in
1258 umgekehrter Reihenfolge durchgeführt und somit rückgängig gemacht werden.

1259 Hinweis: Es MUSS weiters darauf geachtet werden, dass diese ein derart ersetztes PDF-
1260 Steuerzeichen – die sogenannte „escape-sequence“ – niemals geteilt wird. Ein alleine
1261 stehender „\“ vor dem Ende des Bereichs einer Aussparung würde den Abschluss des Bereichs
1262 der Aussparung (End-Delimiter, im PDF mit „)“ dargestellt) „maskieren“ und damit das
1263 Dokument unbrauchbar machen.

1264 **4.5.5 Anwendungshinweis zur Verifikation**

1265 Zur Verifikation von derart signierten Dokumenten MUSS reziprok zu der in dieser Spezifikation
1266 festgelegten Vorgehensweise verfahren werden. Zusätzlich werden die folgenden
1267 Anwendungshinweise gegeben.

1268 Gegeben sei ein unter Anwendung der hier spezifizierten Signaturmethode textuell signiertes
1269 PDF-Dokument. Die Applikation MUSS aus der in der Signatur-Repräsentation enthaltenen
1270 Methoden-Kennung das korrekte Signaturverfahren bestimmen und somit das adäquate
1271 Verifikationsverfahren anwenden.

1272 Die Vorgehensweise der Verifikation im Überblick:

- 1273 1. Aus dem zu verifizierenden PDF-Dokument muss der zur Signatur gehörende Incremental
1274 Update Block – beinhaltet das relevante EGIZ-Dictionary – extrahiert werden.
- 1275 2. Über die im EGIZ-Dictionary eingetragenen Byte-Ranges werden die Aussparungen der
1276 binären PDF-Signatur ermittelt.
- 1277 3. Aus den ermittelten Aussparung werden die Werte extrahiert und gemäß den Vorgaben
1278 dieser Art (Methode) von binären PDF-Signatur als Daten zur Rekonstruktion der XML-
1279 Signatur interpretiert (d.h. Als Signaturattribute, wie Signaturwert, Signaturzeitpunkt, etc.,
1280 bzw. als Signaturparameter sowie Kennzeichnung des angewandten Signaturparameter-
1281 Profils). Diese Daten werden gem. den Vorgaben dieser Art (Methode) von binären PDF-
1282 Signatur behandelt, d.h. zusammengeführt und kodiert, und interpretiert.
- 1283 4. Der Wertebereich der Aussparungen wird gem. den Vorgaben dieser Art (Methode) von
1284 binären PDF-Signatur mit den definierten Füllzeichen (Default-Werten) aufgefüllt werden.
1285 Das danach resultierende PDF-Dokument repräsentiert das binär signierte PDF-
1286 Dokument; dies entspricht dem signierten Datenstrom.
- 1287 5. Die dem signierten PDF-Dokument hinterlegte XML-Signatur wird anhand der aus dem
1288 EGIZ-Dictionary gewonnenen Daten (siehe vorherigen Schritt) unter Berücksichtigung des
1289 jeweiligen Signaturparameter-Profiles, bzw. unter Anwendung des damit festgelegten XML-
1290 Signaturlayouts, rekonstruiert.
- 1291 6. Die rekonstruierte XML-Signatur wird verifiziert.

1292 **5 Definierte Signaturparameter**

1293 Die Signaturparameter nehmen auf Spezialitäten von Signaturerstellungskomponenten
1294 Rücksicht. Vor allem werden damit das Layout und die variablen Elemente der damit erstellten
1295 XML-Signaturen im Rahmen der Signatur-Repräsentation (Signaturblock) zum Ausdruck
1296 gebracht. Dies ist besonders für den Fall der Signaturrekonstruktion auf Basis eines
1297 Papierausdruckes erforderlich (siehe auch allgemeine Erläuterung in Abschnitt 2.2).

1298 Die vorliegende Spezifikation berücksichtigt Spezialitäten einiger Signaturerstellungskomponenten und enthält daher die Definition der folgenden Signaturparameter-Profile, die wie
1299 folgt in Implementierungen unterstützt werden MÜSSEN:
1300

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	DEPRECATED	EMPFOHLEN	NICHT EMPFOHLEN
etsi-moc-1.1	EMPFOHLEN	MUSS	MUSS
etsi-bka-atrust-1.0	EMPFOHLEN	MUSS	MUSS

1301 Eine spezifikationskonforme Umsetzung MUSS die in der obigen Tabelle definierten
1302 Signaturparameterprofile, gemäß den definierten Prioritäten, implementieren.

1303 Jede Implementierung MUSS für die damit erzeugbaren Signaturparameterprofile sowohl die
1304 Signaturerstellung als auch die Signaturverifikation realisieren.

1305 **5.1 Default Signaturparameter-Profil**

1306 **5.1.1 Charakteristik**

1307 *Parameter-Kennzeichnung:* keine Kennzeichnung; es werden keine Parameter
1308 angeführt

1309 *Signaturerstellungskomponente:* für alle Signaturerstellungskomponenten
1310 gem. Spezifikation MOA-SS [7]

1311 *Einschränkungen bzgl. Signaturmethoden:* keine, kann mit allen Signaturmethoden verwendet werden
1312

1313 *Anwendbarkeit:* EMPFOHLEN

1314 5.1.2 Signaturparameter

1315 Für dieses Signaturparameter-Profil KÖNNEN Signaturparameter zur Spezifikation von
1316 Signatur-Suite und Hashalgorithmen (SIGDEV_SPEC) verwendet werden um von den
1317 Standardeinstellungen abweichende Werte zu repräsentieren.

1318 5.1.3 Signaturlayout

1319 Werden keine Signaturparameter angegeben, so MUSS die erstellte Signatur den Vorgaben
1320 einer MOA-SS Signatur [7] folgen und DARF NICHT variable (zeitabhängige oder zufällige)
1321 Werte enthalten.

1322 Eine XML-Signatur, die diesem Parameter-Profil entspricht, MUSS dem folgenden Layout
1323 entsprechen:

```
1324 <dsig:Signature Id="signature-1-1" xmlns:dsig="http://www.w3.org/2000/09/xmldsig#">
1325   <dsig:SignedInfo xmlns:dsig="http://www.w3.org/2000/09/xmldsig#">
1326     <dsig:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-
1327     20010315"/>
1328     <dsig:SignatureMethod Algorithm="ALGORITHM"/>
1329     <dsig:Reference Id="reference-1-1" URI="REFERENCE">
1330       TRANSFORMS
1331       <dsig:DigestMethod Algorithm="DATADIGESTMETHOD"/>
1332       <dsig:DigestValue>DIGESTVALUESIGNEDDATA</dsig:DigestValue>
1333     </dsig:Reference>
1334     <dsig:Reference Type="http://uri.etsi.org/01903/v1.1.1#SignedProperties"
1335     URI="#xmlns(etsi=http://uri.etsi.org/01903/v1.1.1%23)%20xpointer(id('etsi-signed-1-
1336     1')/child::etsi:QualifyingProperties/child::etsi:SignedProperties)">
1337       <dsig:DigestMethod Algorithm="PROPERTIESDIGESTMETHOD"/>
1338       <dsig:DigestValue>DIGESTVALUESIGNEDPROPERTIES</dsig:DigestValue>
1339     </dsig:Reference>
1340   </dsig:SignedInfo>
1341   <dsig:SignatureValue>SIGNATUREVALUE</dsig:SignatureValue>
1342   <dsig:KeyInfo>
1343     <dsig:X509Data>
1344       <dsig:X509Certificate>X509CERTIFICATE</dsig:X509Certificate>
1345     </dsig:X509Data>
1346   </dsig:KeyInfo>
1347   DSIGOBJECT
1348   <dsig:Object Id="etsi-signed-1-1">
1349     <etsi:QualifyingProperties Target="#signature-1-1"
1350     xmlns:etsi="http://uri.etsi.org/01903/v1.1.1#">
1351       <etsi:SignedProperties xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
1352       xmlns:etsi="http://uri.etsi.org/01903/v1.1.1#">
1353         <etsi:SignedSignatureProperties>
1354           <etsi:SigningTime>SIGNINGTIME</etsi:SigningTime>
1355           <etsi:SigningCertificate>
1356             <etsi:Cert>
1357               <etsi:CertDigest>
1358                 <etsi:DigestMethod Algorithm="CERTIFICATEDIGESTMETHOD"/>
1359                 <etsi:DigestValue>DIGESTVALUEX509CERTIFICATE</etsi:DigestValue>
1360               </etsi:CertDigest>
1361               <etsi:IssuerSerial>
1362                 <dsig:X509IssuerName>X509ISSUENAME</dsig:X509IssuerName>
1363                 <dsig:X509SerialNumber>X509SERIALNUMBER</dsig:X509SerialNumber>
1364               </etsi:IssuerSerial>
1365             </etsi:Cert>
1366           </etsi:SigningCertificate>
1367           <etsi:SignaturePolicyIdentifier>
1368             <etsi:SignaturePolicyImplied/>
1369           </etsi:SignaturePolicyIdentifier>
1370         </etsi:SignedSignatureProperties>
```

```
1371     <etsi:SignedDataObjectProperties>  
1372     <etsi:DataObjectFormat ObjectReference="#reference-1-1">  
1373     <etsi:MimeType>MIMETYPE</etsi:MimeType>  
1374     </etsi:DataObjectFormat>  
1375     </etsi:SignedDataObjectProperties>  
1376     </etsi:SignedProperties>  
1377     </etsi:QualifyingProperties>  
1378 </dsig:Object>  
1379 </dsig:Signature>
```

1380 Die vom Layout vorgesehenen Variabilitäten werden in Großbuchstaben und umrandet
1381 ausgewiesen. Die nachfolgenden Abschnitte definieren diese im Detail.

1382 Dieses Profil legt durch das damit festgelegte XML-Signaturlayout die folgenden wesentlichen
1383 XML-Signatur-Eigenschaften implizit fest:

1384 1. Kanonisierungsmethode:

1385 <http://www.w3.org/TR/2001/REC-xml-c14n-20010315>

1386 2. Digest-Methoden: Wird im Layout angegeben. Sonst mit dem Standardwert:

1387 <http://www.w3.org/2000/09/xmlldsig#sha1>

1388 3. Qualifying Properties: nach ETSI (<http://uri.etsi.org/01903/v1.1.1#>)

1389 5.1.3.1 ALGORITHM

1390 Legt den Signaturalgorithmus fest. Wenn dieser nicht im Signaturparameter definiert ist, werden
1391 folgende Standardwerte verwendet (in Abhängigkeit der Eigenschaften des Signatur-
1392 Schlüssels):

1393 • für ECDSA-Schlüssel: <http://www.w3.org/2000/09/xmlldsig#rsa-sha1>

1394 • für RSA-Schlüssel: <http://www.w3.org/2000/09/xmlldsig#rsa-sha1>

1395 Der zum Signaturschlüssel passende Identifikator (URI) MUSS im XML-Signaturlayout anstelle
1396 der Variable **ALGORITHM** eingesetzt werden.

1397 5.1.3.2 REFERENCE

1398 Wird eine Signatur-Methode angewandt, die eine enveloping XML-Signatur erzeugt, so MUSS
1399 das folgende Fragment anstelle der Variable **REFERENCE** im XML-Signaturlayout eingesetzt
1400 werden:

```
1401 #xmlns(etsi=http://uri.etsi.org/01903/v1.1.1%23)%20xpointer(id('ets  
1402 i-signed-1-1')/child::etsi:QualifyingProperties/child::etsi:Sig  
1403 nedProperties)
```

1404 Wird eine Signatur-Methode angewandt, die eine detached XML-Signatur erzeugt, so MUSS
1405 das folgende Fragment anstelle der Variable **REFERENCE** im XML-Signaturlayout eingesetzt
1406 werden:

```
1407 urn:Document
```

1408 5.1.3.3 TRANSFORMS

1409 Wird eine Signatur-Methode angewandt, die eine enveloping XML-Signatur erzeugt, so MUSS
1410 das folgende Fragment anstelle der Variable **TRANSFORMS** im XML-Signaturlayout eingesetzt
1411 werden:

```
1412 <dsig:Transforms>  
1413 <dsig:Transform Algorithm="http://www.w3.org/2000/09/xmlldsig#base64"/>  
1414 </dsig:Transforms>
```

1415 Wird eine Signatur-Methode angewandt, die eine detached XML-Signatur erwirkt, so darf kein
1416 Transformationspfad eingefügt werden. In diesem Fall DARF ein Transformationspfad NICHT

1417 im XML-Signaturlayout eingesetzt werden. Die Variable **TRANSFORMS** im XML-Signaturlayout
1418 MUSS ersatzlos entfernt werden.

1419 **5.1.3.4 DIGESTVALUESIGNEDDATA**

1420 Anstelle der Variable **DIGESTVALUESIGNEDDATA** MUSS im XML-Signaturlayout der Hash-
1421 Wert der ersten XML-Signaturreferenz eingesetzt werden.

1422 **5.1.3.5 DATADIGESTMETHOD**

1423 Anstelle der Variable **DATADIGESTMETHOD** MUSS im XML-Signaturlayout die Hashmethode
1424 (URI) der ersten XML-Signaturreferenz eingesetzt werden.

1425 **5.1.3.6 DIGESTVALUESIGNEDPROPERTIES**

1426 Anstelle der Variable **DIGESTVALUESIGNEDPROPERTIES** MUSS im XML-Signaturlayout der
1427 Hash-Wert der zweiten XML-Signaturreferenz eingesetzt werden.

1428 **5.1.3.7 PROPERTIESDIGESTMETHOD**

1429 Anstelle der Variable **PROPERTYDIGESTMETHOD** MUSS im XML-Signaturlayout die
1430 Hashmethode (URI) der zweiten XML-Signaturreferenz eingesetzt werden.

1431 **5.1.3.8 SIGNATUREVALUE**

1432 Anstelle der Variable **SIGNATUREVALUE** MUSS im XML-Signaturlayout der Signaturwert
1433 eingesetzt werden.

1434 **5.1.3.9 X509CERTIFICATE**

1435 Anstelle der Variable **X509CERTIFICATE** MUSS im XML-Signaturlayout das Base64-kodierte
1436 Signaturzertifikat (X509-Zertifikat, kodiert gem. Abschnitt 4.4.4 in [4]) eingesetzt werden.

1437 **5.1.3.10 DSIGOBJECT**

1438 Wird eine Signatur-Methode angewandt, die eine enveloping XML-Signatur erzeugt, so MUSS
1439 das folgende Fragment anstelle der Variable **DSIGOBJECT** im XML-Signaturlayout eingesetzt
1440 werden:

```
1441     <dsig:Object Id="signed-data-1-1-1">  
1442         <Base64Content>BASE64CONTENT</Base64Content>  
1443     </dsig:Object>
```

1444 In diesem Fragment MUSS anstelle der Variable **BASE64CONTENT** der Base64-kodierte, zu
1445 signierende Datenstrom (gem. angewandter Signaturmethode) eingesetzt werden.

1446 Wird eine Signatur-Methode angewandt, die eine detached XML-Signatur erzeugt, so ist kein
1447 XML-Signaturobjekt für die signierten Daten notwendig. Die Variable **DSIGOBJECT** im XML-
1448 Signaturlayout MUSS ersatzlos entfernt werden.

1449 **5.1.3.11 SIGNINGTIME**

1450 Anstelle der Variable **SIGNINGTIME** MUSS der Signaturzeitpunkt gem. [5] eingesetzt werden.

1451 **5.1.3.12 DIGESTVALUEX509CERTIFICATE**

1452 Anstelle der Variable **DIGESTVALUEX509CERTIFICATE** MUSS der Hash-Wert des
1453 Signaturzertifikates (lt. [5]) eingesetzt werden.

1454 **5.1.3.13 CERTIFICATEDIGESTMETHOD**

1455 Anstelle der Variable `CERTIFICATEDIGESTMETHOD` MUSS im XML-Signaturlayout die
1456 Hashmethode (URI) des Zertifikatdigests eingesetzt werden.

1457 **5.1.3.14 X509ISSUERNAME**

1458 Anstelle der Variable `X509ISSUERNAME` MUSS der Name des Ausstellers des
1459 Signaturzertifikates (lt. [5]) eingesetzt werden.

1460 **5.1.3.15 X509SERIALNUMBER**

1461 Anstelle der Variable `X509SERIALNUMBER` MUSS die Seriennummer des Signaturzertifikates
1462 (lt. [5]) eingesetzt werden.

1463 **5.1.3.16 MIMETYPE**

1464 Anstelle der Variable `MIMETYPE` MUSS der MIME-Type der zu signierenden Daten eingesetzt
1465 werden. Der MIME-Type ist von der angewandten Signaturmethode abhängig und MUSS im
1466 Zuge der Definition der Signaturmethode festgelegt werden.

1467 **5.2 Default Signaturparameter-Profil für BKU**

1468 **5.2.1 Charakteristik**

1469 *Parameter-Kennzeichnung:* keine Kennzeichnung; es werden zwar explizite
1470 Signaturparameter eingeführt, dieses Profil hat jedoch
1471 keine eigene Kennzeichnung. Es werden nur die Parameter
1472 in der Signatur-Repräsentation (Signaturblock) eingefügt.

1473 *Signaturerstellungskomponente:* dieses Profil MUSS für die Generallizenz der
1474 Bürgerkartenumgebung (IT-Solution trustDesk-Basic), bis
1475 zur Version 2.7.4. angewendet werden.

1476 *Einschränkungen bzgl. Signatur-*
1477 *methoden:* dieses Profil MUSS mit Signaturmethoden verwendet
1478 werden, die eine enveloping Signatur erzeugen; diese sind:
1479 urn:pdfsigfilter:bka.gv.at:binaer:v1.0.0 und
1480 urn:pdfsigfilter:bka.gv.at:text:v1.0.0

1481 *Verwendbarkeit:* Dieses Signaturparameter-Profil DARF NICHT mehr zur
1482 Anwendung gelangen (deprecated). Es wurde durch das
1483 ets-bka-1.0 Signaturparameter-Profil ersetzt.

1484 **5.2.2 Signaturparameter**

1485 Für dieses Signaturparameter-Profil wurden Signaturparameter definiert. Diese MÜSSEN ohne
1486 vorangestellte Kennzeichnung in die Signatur-Repräsentation eingefügt werden.

1487 Für die Formulierung der Signaturparameter MUSS die Definition gem. Abschnitt 2.2. Es gilt bei
1488 diesem Signaturparameter-Profil jedoch die Maßgabe, dass das Feld `<SIGDEV_PROF>` leer
1489 bleiben MUSS.

1490 Die Signaturparameter dieses Profils repräsentieren 5 Einzelwerte. Dabei MUSS als Parameter
1491 Teil 1 (Feld PARAM_L1 der Signaturparameter, siehe 2.2) als konstanter Präfix für alle 5
1492 Einzelwerte herangezogen werden. Der Parameter Teil 2 (Feld PARAM_L2 der
1493 Signaturparameter, siehe 2.2) enthält selbst 5 Einzelwerte, die jeweils mit einem Bindestrich
1494 separiert werden müssen. Es gilt für Parameter Teil 2 folgende Ausformung (in Ergänzung zur
1495 Definition aus Abschnitt 2.2):

```
1496 <PARAM_L2> ::= <WERT_1>"-"<WERT_2>"-"<WERT_3>"-"<WERT_4>"-"<WERT_5>  
1497 <WERT_1> ::= 1*<CHAR>  
1498 <WERT_2> ::= 1*<CHAR>  
1499 <WERT_3> ::= 1*<CHAR>  
1500 <WERT_4> ::= 1*<CHAR>  
1501 <WERT_5> ::= 1*<CHAR>
```

1502 Daraus MÜSSEN folgende Einzelwerte gebildet werden:

```
1503 <ParamSigID> ::= <PARAM_L1>"-"<WERT_1>  
1504 <ParamSigDataRef> ::= "0-"<PARAM_L1>"-"<WERT_2>  
1505 <ParamSigDataObjURI> ::= "0-"<PARAM_L1>"-"<WERT_3>  
1506 <ParamEtsiDataRef> ::= "0-"<PARAM_L1>"-"<WERT_4>  
1507 <ParamEtsiDataObjURI> ::= "0-"<PARAM_L1>"-"<WERT_5>
```

1508 Dieser Werte werden mehrfach im Signaturlayout referenziert und verwendet. Wird eine
1509 Signatur-Methode angewandt, die eine detached XML-Signatur erzeugt, so wird der Wert
1510 <ParamSigDataObjURI> nicht benötigt. Die Signaturparameter lassen die Bildung dieses
1511 Wertes jedoch dennoch zu.

1512 Die Verwendung dieser Einzelwerte wird in Abschnitt 5.2.3 festgelegt.

1513 Nachfolgend ein Beispiel zur Bildung der Einzelwerte auf Basis übergebener Signaturparameter
1514 (spezifisch für das vorliegende Signaturparameter-Profil):

1515 Signaturparameter lt. Signaturrepräsentation:

```
1516 @1200412799-27800484@23524-22018-26789-24095-30271
```

1517 Daraus ergeben sich:

```
1518 <ParamSigID> = 1200412799-27800484-23524  
1519 <ParamSigDataRef> = 0-1200412799-27800484-22018  
1520 <ParamSigDataObjURI> = 0-1200412799-27800484-26789  
1521 <ParamEtsiDataRef> = 0-1200412799-27800484-24095  
1522 <ParamEtsiDataObjURI> = 0-1200412799-27800484-30271
```

1523 5.2.3 Signaturlayout

1524 Die Signaturen einer Bürgerkartenumgebung enthalten zeitabhängige Varianzen. Diese sind vor
1525 allem eine Reihe von XML-Attributen (ID-Attribute) die zur Referenzierung von XML-
1526 Elementen/-Knoten herangezogen werden. Diese variablen Attribute MÜSSEN in Form der
1527 Signaturparameter innerhalb der Signatur-Repräsentation verwaltet und bei der Rekonstruktion
1528 der Signatur entsprechend berücksichtigt werden.

1529 Eine XML-Signatur, die diesem Parameter-Profil entspricht, MUSS dem folgenden Layout
1530 entsprechen:

```
1531 <dsig:Signature xmlns:dsig="http://www.w3.org/2000/09/xmldsig#" Id="signature-SIGID">
1532   <dsig:SignedInfo>
1533     <dsig:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-
1534     20010315"/>
1535     <dsig:SignatureMethod Algorithm="ALGORITHM"/>
1536     <dsig:Reference Id="signed-data-reference-SIGDATAREF" URI="#signed-data-object-
1537     SIGDATAOBJURI">
1538       <dsig:Transforms>
1539         <dsig:Transform Algorithm="http://www.w3.org/2002/06/xmldsig-filter2">
1540           <xpf:XPath xmlns:xpf="http://www.w3.org/2002/06/xmldsig-filter2"
1541           Filter="intersect">id('signed-data-object-SIGDATAOBJURI')/node()</xpf:XPath>
1542         </dsig:Transform>
1543         <dsig:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#base64"/>
1544       </dsig:Transforms>
1545       <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
1546       <dsig:DigestValue>DIGESTVALUESIGNEDDATA</dsig:DigestValue>
1547     </dsig:Reference>
1548     <dsig:Reference Id="etsi-data-reference-ETSIDATAREF"
1549     Type="http://uri.etsi.org/01903/v1.1.1#SignedProperties" URI="#etsi-data-object-
1550     ETSIDATAOBJURI">
1551       <dsig:Transforms>
1552         <dsig:Transform Algorithm="http://www.w3.org/2002/06/xmldsig-filter2">
1553           <xpf:XPath xmlns:xpf="http://www.w3.org/2002/06/xmldsig-filter2"
1554           Filter="intersect">id('etsi-data-object-ETSIDATAOBJURI')/node()</xpf:XPath>
1555         </dsig:Transform>
1556       </dsig:Transforms>
1557       <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
1558       <dsig:DigestValue>DIGESTVALUESIGNEDPROPERTIES</dsig:DigestValue>
1559     </dsig:Reference>
1560   </dsig:SignedInfo>
1561   <dsig:SignatureValue>SIGNATUREVALUE</dsig:SignatureValue>
1562   <dsig:KeyInfo>
1563     <dsig:X509Data>
1564       <dsig:X509Certificate>X509CERTIFICATE</dsig:X509Certificate>
1565     </dsig:X509Data>
1566   </dsig:KeyInfo>
1567   <dsig:Object Id="signed-data-object-SIGDATAOBJURI">
1568     <sl:Base64Content>BASE64CONTENT</sl:Base64Content>
1569   </dsig:Object>
1570   <dsig:Object Id="etsi-data-object-ETSIDATAOBJURI">
1571     <etsi:QualifyingProperties xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
1572     xmlns:etsi="http://uri.etsi.org/01903/v1.1.1#" Target="#signature-SIGID">
1573       <etsi:SignedProperties>
1574         <etsi:SignedSignatureProperties>
1575           <etsi:SigningTime>SIGNINGTIME</etsi:SigningTime>
```

```
1576 <etsi:SigningCertificate>
1577 <etsi:Cert>
1578 <etsi:CertDigest>
1579 <etsi:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
1580 <etsi:DigestValue>DIGESTVALUEX509CERTIFICATE</etsi:DigestValue>
1581 </etsi:CertDigest>
1582 <etsi:IssuerSerial>
1583 <dsig:X509IssuerName>X509ISSUERNAME</dsig:X509IssuerName>
1584 <dsig:X509SerialNumber>X509SERIALNUMBER</dsig:X509SerialNumber>
1585 </etsi:IssuerSerial>
1586 </etsi:Cert>
1587 </etsi:SigningCertificate>
1588 <etsi:SignaturePolicyIdentifier>
1589 <etsi:SignaturePolicyImplied/>
1590 </etsi:SignaturePolicyIdentifier>
1591 </etsi:SignedSignatureProperties>
1592 <etsi:SignedDataObjectProperties>
1593 <etsi:DataObjectFormat ObjectReference="#signed-data-reference-SIGDATAREF">
1594 <etsi:MimeType>text/plain</etsi:MimeType>
1595 </etsi:DataObjectFormat>
1596 </etsi:SignedDataObjectProperties>
1597 </etsi:SignedProperties>
1598 </etsi:QualifyingProperties>
1599 </dsig:Object>
1600 </dsig:Signature>
```

1601 Die vom Layout vorgesehenen Variabilitäten (Variablen) werden in Großbuchstaben und
1602 umrandet ausgewiesen. Die nachfolgenden Abschnitte definieren diese im Detail.

1603 Dieses Profil legt durch das damit festgelegte XML-Signaturlayout die folgenden wesentlichen
1604 XML-Signatur-Eigenschaften implizit fest:

- 1605 1. Kanonisierungsmethode: <http://www.w3.org/TR/2001/REC-xml-c14n-20010315>
- 1606 2. Digest-Methode: <http://www.w3.org/2000/09/xmldsig#sha1>
- 1607 3. Qualifying Properties: nach ETSI (<http://uri.etsi.org/01903/v1.1.1#>)

1608 5.2.3.1 SIGID

1609 Anstelle der Variable **SIGID** MUSS im XML-Signaturlayout der aus den Signaturparametern
1610 zusammengesetzte Wert <ParamSigID> eingesetzt werden.

1611 5.2.3.2 ALGORITHM

1612 Legt den Signaturalgorithmus fest. Es MUSS einer der folgenden Signaturalgorithmen
1613 verwendet werden, in Abhängigkeit der Eigenschaften des Signatur-Schlüssels:

- 1614 • für ECDSA-Schlüssel: <http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha1>
- 1615 • für RSA-Schlüssel: <http://www.w3.org/2000/09/xmldsig#rsa-sha1>

1616 Der zum Signaturschlüssel passende Identifikator (URI) MUSS im XML-Signaturlayout anstelle
1617 der Variable **ALGORITHM** eingesetzt werden.

1618 5.2.3.3 SIGDATAREF

1619 Anstelle der Variable **SIGDATAREF** MUSS im XML-Signaturlayout der aus den
1620 Signaturparametern zusammengesetzte Wert <ParamSigDataRef> eingesetzt werden.

1621 5.2.3.4 SIGDATAOBJURI

1622 Anstelle der Variable **SIGDATAOBJURI** MUSS im XML-Signaturlayout der aus den
1623 Signaturparametern zusammengesetzte Wert <ParamSigDataObjURI> eingesetzt werden.

1624 **5.2.3.5 DIGESTVALUESIGNEDDATA**

1625 Anstelle der Variable `DIGESTVALUESIGNEDDATA` MUSS im XML-Signaturlayout der Hash-
1626 Wert der ersten XML-Signaturreferenz eingesetzt werden.

1627 **5.2.3.6 ETSIDATAREF**

1628 Anstelle der Variable `ETSIDATAREF` MUSS im XML-Signaturlayout der aus den
1629 Signaturparametern zusammengesetzte Wert `<ParamEtsiDataRef>` eingesetzt werden.

1630 **5.2.3.7 ETSIDATAOBJURI**

1631 Anstelle der Variable `ETSIDATAOBJURI` MUSS im XML-Signaturlayout der aus den
1632 Signaturparametern zusammengesetzte Wert `<ParamEtsiDataObjURI>` eingesetzt werden.

1633 **5.2.3.8 DIGESTVALUESIGNEDPROPERTIES**

1634 Anstelle der Variable `DIGESTVALUESIGNEDPROPERTIES` MUSS im XML-Signaturlayout der
1635 Hash-Wert der zweiten XML-Signaturreferenz eingesetzt werden.

1636 **5.2.3.9 SIGNATUREVALUE**

1637 Anstelle der Variable `SIGNATUREVALUE` MUSS im XML-Signaturlayout der Signaturwert
1638 eingesetzt werden.

1639 **5.2.3.10 X509CERTIFICATE**

1640 Anstelle der Variable `X509CERTIFICATE` MUSS im XML-Signaturlayout das Base64-kodierte
1641 Signaturzertifikat (X509-Zertifikat, kodiert gem. Abschnitt 4.4.4 in [4]) eingesetzt werden.

1642 **5.2.3.11 DSIGOBJECT**

1643 Wird eine Signatur-Methode angewandt, die eine enveloping XML-Signatur erwirkt, so MUSS
1644 das folgende Fragment anstelle der Variable `DSIGOBJECT` im XML-Signaturlayout eingesetzt
1645 werden:

```
1646 <dsig:Object Id="signed-data-object-SIGDATAOBJURI">  
1647 <dsig:Base64Content>BASE64CONTENT</dsig:Base64Content>  
1648 </dsig:Object>
```

1649 In diesem Fragment MUSS anstelle der Variable `SIGDATAOBJURI` der aus den
1650 Signaturparametern zusammengesetzte Wert `<ParamSigDataObjURI>` eingesetzt werden.

1651 In diesem Fragment MUSS anstelle der Variable `BASE64CONTENT` der Base64-kodierte, zu
1652 signierende Datenstrom (gem. angewandter Signaturmethode) eingesetzt werden.

1653 Wird eine Signatur-Methode angewandt, die eine detached XML-Signatur erwirkt, so ist kein
1654 XML-Signaturobjekt für die signierten Daten notwendig. Die Variable `DSIGOBJECT` im XML-
1655 Signaturlayout MUSS ersatzlos entfernt werden.

1656 **5.2.3.12 SIGNINGTIME**

1657 Anstelle der Variable `SIGNINGTIME` MUSS der Signaturzeitpunkt gem. [5] eingesetzt werden.

1658 **5.2.3.13 DIGESTVALUEX509CERTIFICATE**

1659 Anstelle der Variable `DIGESTVALUEX509CERTIFICATE` MUSS der Hash-Wert des
1660 Signaturzertifikates (lt. [5]) eingesetzt werden.

1661 **5.2.3.14 X509ISSUERNAME**

1662 Anstelle der Variable **X509ISSUERNAME** MUSS der Name des Ausstellers des
1663 Signaturzertifikates (lt. [5]) eingesetzt werden.

1664 **5.2.3.15 X509SERIALNUMBER**

1665 Anstelle der Variable **X509SERIALNUMBER** MUSS die Seriennummer des Signaturzertifikates
1666 (lt. [5]) eingesetzt werden.

1667 **5.3 Signaturparameter-Profil etsi-bka-1.0**

1668 **5.3.1 Charakteristik**

1669 *Parameter-Kennzeichnung:* etsi-bka-1.0

1670 *Signaturerstellungskomponente:* dieses Profil MUSS für die Generallizenz der
1671 Bürgerkartenumgebung (IT-Solution trustDesk-Basic), ab
1672 Version 2.7.5 angewendet werden.

1673 *Einschränkungen bzgl. Signatur-*
1674 *methoden:* keine, kann mit allen Signaturmethoden verwendet werden

1675 *Verwendbarkeit:* EMPFOHLEN

1676 **5.3.2 Signaturparameter**

1677 Für dieses Signaturparameter-Profil wurden Signaturparameter definiert. Diese MÜSSEN nach
1678 der vorangestellten Kennzeichnung des Profils in die Signatur-Repräsentation eingefügt
1679 werden.

1680 Für die Formulierung der Signaturparameter MUSS die Definition gem. Abschnitt 2.2 angewandt
1681 werden. Es gilt bei diesem Signaturparameter-Profil jedoch die Maßgabe, dass das Feld
1682 <SIGDEV_PROF> den Wert etsi-bka-1.0 haben MUSS.

1683 Die Signaturparameter dieses Profils repräsentieren 5 Einzelwerte. Dabei MUSS als Parameter
1684 Teil 1 (Feld PARAM_L1 der Signaturparameter, siehe 2.2) als konstanter Präfix für alle 5
1685 Einzelwerte herangezogen werden. Der Parameter Teil 2 (Feld PARAM_L2 der
1686 Signaturparameter, siehe 2.2) enthält selbst 5 Einzelwerte, die jeweils mit einem Bindestrich
1687 separiert werden müssen. Es gilt für Parameter Teil 2 folgende Ausformung (in Ergänzung zur
1688 Definition aus Abschnitt 2.2):

1689 <PARAM_L2> ::= <WERT_1>"-"<WERT_2>"-"<WERT_3>"-"<WERT_4>"-"<WERT_5>
1690 <WERT_1> ::= <CHAR>
1691 <WERT_2> ::= <CHAR>
1692 <WERT_3> ::= <CHAR>
1693 <WERT_4> ::= <CHAR>
1694 <WERT_5> ::= <CHAR>

1695 Daraus MÜSSEN folgende Einzelwerte gebildet werden:

1696 <ParamSigID> ::= <PARAM_L1>"-"<WERT_1>
1697 <ParamSigDataRef> ::= "0-"<PARAM_L1>"-"<WERT_2>
1698 <ParamSigDataObjURI> ::= "0-"<PARAM_L1>"-"<WERT_3>
1699 <ParamEtsiDataRef> ::= "0-"<PARAM_L1>"-"<WERT_4>
1700 <ParamEtsiDataObjURI> ::= "0-"<PARAM_L1>"-"<WERT_5>

1701 Dieser Werte werden mehrfach im Signaturlayout referenziert und verwendet. Wird eine
1702 Signatur-Methode angewandt, die eine detached XML-Signatur erzeugt, so wird der Wert
1703 <ParamSigDataObjURI> nicht benötigt. Die Signaturparameter lassen die Bildung dieses
1704 Wertes jedoch dennoch zu.

1705 Die Verwendung dieser Einzelwerte wird in Abschnitt 5.2.3 festgelegt.

1706 Nachfolgend ein Beispiel zur Bildung der Einzelwerte auf Basis übergebener Signaturparameter
1707 (spezifisch für das vorliegende Signaturparameter-Profil):

1708 Signaturparameter lt. Signaturrepräsentation:

1709 etsi-bka-1.0@1200412799-27800484@23524-22018-0-24095-30271

1710 Daraus ergeben sich:

1711 <ParamSigID> = 1200412799-27800484-23524
1712 <ParamSigDataRef> = 0-1200412799-27800484-22018
1713 <ParamSigDataObjURI> = 0-1200412799-27800484-0
1714 <ParamEtsiDataRef> = 0-1200412799-27800484-24095
1715 <ParamEtsiDataObjURI> = 0-1200412799-27800484-30271

1716 5.3.3 Signaturlayout

1717 Die Signaturen einer Bürgerkartenumgebung enthalten zeitabhängige Varianzen. Diese sind vor
1718 allem eine Reihe von XML-Attributen (ID-Attribute) die zur Referenzierung von XML-
1719 Elementen/-Knoten herangezogen werden. Diese variablen Attribute MÜSSEN in Form der
1720 Signaturparameter innerhalb der Signatur-Repräsentation verwaltet und bei der Rekonstruktion
1721 der Signatur entsprechend berücksichtigt werden.

1722 Eine XML-Signatur, die diesem Parameter-Profil entspricht, MUSS dem folgenden Layout
1723 entsprechen:

```
1724 <dsig:Signature xmlns:dsig="http://www.w3.org/2000/09/xmldsig#" Id="signature-SIGID">  
1725   <dsig:SignedInfo>  
1726     <dsig:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-  
1727     20010315"/>  
1728     <dsig:SignatureMethod Algorithm="ALGORITHM"/>  
1729     <dsig:Reference Id="signed-data-reference-SIGDATAREF" URI="REFERENCE">  
1730       TRANSFORMS  
1731       <dsig:DigestMethod Algorithm="DATADIGESTMETHOD"/>  
1732       <dsig:DigestValue>DIGESTVALUESIGNEDDATA</dsig:DigestValue>  
1733     </dsig:Reference>  
1734     <dsig:Reference Id="etsi-data-reference-ETSIDATAREF"  
1735     Type="http://uri.etsi.org/01903/v1.1.1#SignedProperties"  
1736     URI="#xmlns(etsi=http://uri.etsi.org/01903/v1.1.1%23)%20xpointer(id('etsi-data-object-  
1737     ETSIDATAOBJURI')/child::etsi:QualifyingProperties/child::etsi:SignedProperties)">  
1738       <dsig:DigestMethod Algorithm="PROPERTYDIGESTMETHOD"/>  
1739       <dsig:DigestValue>DIGESTVALUESIGNEDPROPERTIES</dsig:DigestValue>  
1740     </dsig:Reference>  
1741   </dsig:SignedInfo>  
1742   <dsig:SignatureValue>SIGNATUREVALUE</dsig:SignatureValue>  
1743   <dsig:KeyInfo>  
1744     <dsig:X509Data>  
1745       <dsig:X509Certificate>X509CERTIFICATE</dsig:X509Certificate>  
1746     </dsig:X509Data>  
1747   </dsig:KeyInfo>  
1748   DSIGOBJECT  
1749   <dsig:Object Id="etsi-data-object-ETSIDATAOBJURI">  
1750     <etsi:QualifyingProperties xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"  
1751     xmlns:etsi="http://uri.etsi.org/01903/v1.1.1#" Target="#signature-SIGID">  
1752       <etsi:SignedProperties xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"  
1753       xmlns:etsi="http://uri.etsi.org/01903/v1.1.1#">  
1754         <etsi:SignedSignatureProperties>  
1755         <etsi:SigningTime>SIGNINGTIME</etsi:SigningTime>
```

```
1756     <etsi:SigningCertificate>
1757     <etsi:Cert>
1758     <etsi:CertDigest>
1759     <etsi:DigestMethod Algorithm="CERTIFICATEDIGESTMETHOD"/>
1760     <etsi:DigestValue>DIGESTVALUEX509CERTIFICATE</etsi:DigestValue>
1761     </etsi:CertDigest>
1762     <etsi:IssuerSerial>
1763     <dsig:X509IssuerName>X509ISSUERNAME</dsig:X509IssuerName>
1764     <dsig:X509SerialNumber>X509SERIALNUMBER</dsig:X509SerialNumber>
1765     </etsi:IssuerSerial>
1766     </etsi:Cert>
1767     </etsi:SigningCertificate>
1768     <etsi:SignaturePolicyIdentifier>
1769     <etsi:SignaturePolicyImplied/>
1770     </etsi:SignaturePolicyIdentifier>
1771 </etsi:SignedSignatureProperties>
1772 <etsi:SignedDataObjectProperties>
1773     <etsi:DataObjectFormat ObjectReference="#signed-data-reference-SIGDATAREF">
1774     <etsi:MimeType>MIMETYPE</etsi:MimeType>
1775     </etsi:DataObjectFormat>
1776 </etsi:SignedDataObjectProperties>
1777 </etsi:SignedProperties>
1778 </etsi:QualifyingProperties>
1779 </dsig:Object>
1780 </dsig:Signature>
```

1781 Die vom Layout vorgesehenen Variabilitäten (Variablen) werden in Großbuchstaben und
1782 umrandet ausgewiesen. Die nachfolgenden Abschnitte definieren diese im Detail.

1783 Dieses Profil legt durch das damit festgelegte XML-Signaturlayout die folgenden wesentlichen
1784 XML-Signatur-Eigenschaften implizit fest:

- 1785 1. Kanonisierungsmethode: <http://www.w3.org/TR/2001/REC-xml-c14n-20010315>
- 1786 2. Digest-Methode: Wird im Layout angegeben. Sonst mit dem Standardwert:
1787 <http://www.w3.org/2000/09/xmlsig#sha1>
- 1788 3. Qualifying Properties: nach ETSI (<http://uri.etsi.org/01903/v1.1.1#>)

1789 5.3.3.1 SIGID

1790 Anstelle der Variable **SIGID** MUSS im XML-Signaturlayout der aus den Signaturparametern
1791 zusammengesetzte Wert <ParamSigID> eingesetzt werden.

1792 5.3.3.2 ALGORITHM

1793 Legt den Signaturalgorithmus fest. Wenn dieser nicht im Signaturparameter definiert ist, werden
1794 folgende Standardwerte verwendet (in Abhängigkeit der Eigenschaften des Signatur-
1795 Schlüssels):

- 1796 • für ECDSA-Schlüssel: <http://www.w3.org/2001/04/xmlsig-more#ecdsa-sha1>
- 1797 • für RSA-Schlüssel: <http://www.w3.org/2000/09/xmlsig#rsa-sha1>

1798 Der zum Signaturschlüssel passende Identifikator (URI) MUSS im XML-Signaturlayout anstelle
1799 der Variable **ALGORITHM** eingesetzt werden.

1800 5.3.3.3 SIGDATAREF

1801 Anstelle der Variable **SIGDATAREF** MUSS im XML-Signaturlayout der aus den
1802 Signaturparametern zusammengesetzte Wert <ParamSigDataRef> eingesetzt werden.

1803 **5.3.3.4 SIGDATAOBJURI**

1804 Anstelle der Variable `SIGDATAOBJURI` MUSS im XML-Signaturlayout der aus den
1805 Signaturparametern zusammengesetzte Wert `<ParamSigDataObjURI>` eingesetzt werden.

1806 Dieser Wert wird mehrfach im Signaturlayout referenziert und verwendet. Wird eine Signatur-
1807 Methode angewandt, die eine detached XML-Signatur erzeugt, so wird dieser Wert nicht
1808 benötigt. Die Signaturparameter lassen die Bildung dieses Wertes jedoch dennoch zu.

1809 **5.3.3.5 REFERENCE**

1810 Wird eine Signatur-Methode angewandt, die eine enveloping XML-Signatur erwirkt, so MUSS
1811 das folgende Fragment anstelle der Variable `REFERENCE` im XML-Signaturlayout eingesetzt
1812 werden:

```
1813 #signed-data-object-SIGDATAOBJURI
```

1814 In diesem Fragment MUSS anstelle der Variable `SIGDATAOBJURI` der aus den
1815 Signaturparametern zusammengesetzte Wert `<ParamSigDataObjURI>` eingesetzt werden.

1816 Beispiel: `#signed-data-object-0-1155648477-25748375-22389`

1817 Wird eine Signatur-Methode angewandt, die eine detached XML-Signatur erzeugt, so MUSS
1818 das folgende Fragment anstelle der Variable `REFERENCE` im XML-Signaturlayout eingesetzt
1819 werden:

```
1820 urn:Document
```

1821 **5.3.3.6 TRANSFORMS**

1822 Wird eine Signatur-Methode angewandt, die eine enveloping XML-Signatur erzeugt, so MUSS
1823 das folgende Fragment anstelle der Variable `TRANSFORMS` im XML-Signaturlayout eingesetzt
1824 werden:

```
1825 <dsig:Transforms>  
1826   <dsig:Transform Algorithm="http://www.w3.org/2002/06/xmldsig-filter2">  
1827     <xpf:XPath xmlns:xpf="http://www.w3.org/2002/06/xmldsig-filter2"  
1828     Filter="intersect">id('signed-data-object-SIGDATAOBJURI')/node()</xpf:XPath>  
1829   </dsig:Transform>  
1830   <dsig:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#base64"/>  
1831 </dsig:Transforms>
```

1832 In diesem Fragment MUSS anstelle der Variable `SIGDATAOBJURI` der aus den
1833 Signaturparametern zusammengesetzte Wert `<ParamSigDataObjURI>` eingesetzt werden.

1834 Wird eine Signatur-Methode angewandt, die eine detached XML-Signatur erwirkt, so ist kein
1835 Transformationspfad einzufügen. In diesem Fall DARF ein Transformationspfad NICHT im XML-
1836 Signaturlayout eingesetzt werden. Die Variable `TRANSFORMS` im XML-Signaturlayout MUSS
1837 ersatzlos entfernt werden.

1838 **5.3.3.7 DIGESTVALUESIGNEDDATA**

1839 Anstelle der Variable `DIGESTVALUESIGNEDDATA` MUSS im XML-Signaturlayout der Hash-
1840 Wert der ersten XML-Signaturreferenz eingesetzt werden.

1841 **5.3.3.8 DATADIGESTMETHOD**

1842 Anstelle der Variable `DATADIGESTMETHOD` MUSS im XML-Signaturlayout die Hashmethode
1843 (URI) der ersten XML-Signaturreferenz eingesetzt werden.

1844 **5.3.3.9 ETSIDATAREF**

1845 Anstelle der Variable **ETSIDATAREF** MUSS im XML-Signaturlayout der aus den
1846 Signaturparametern zusammengesetzte Wert `<ParamEtsiDataRef>` eingesetzt werden.

1847 **5.3.3.10 ETSIDATAOBJURI**

1848 Anstelle der Variable **ETSIDATAOBJURI** MUSS im XML-Signaturlayout der aus den
1849 Signaturparametern zusammengesetzte Wert `<ParamEtsiDataObjURI>` eingesetzt werden.

1850 **5.3.3.11 DIGESTVALUESIGNEDPROPERTIES**

1851 Anstelle der Variable **DIGESTVALUESIGNEDPROPERTIES** MUSS im XML-Signaturlayout der
1852 Hash-Wert der zweiten XML-Signaturreferenz eingesetzt werden.

1853 **5.3.3.12 PROPERTIESDIGESTMETHOD**

1854 Anstelle der Variable **PROPERTYDIGESTMETHOD** MUSS im XML-Signaturlayout die
1855 Hashmethode (URI) der zweiten XML-Signaturreferenz eingesetzt werden.

1856 **5.3.3.13 SIGNATUREVALUE**

1857 Anstelle der Variable **SIGNATUREVALUE** MUSS im XML-Signaturlayout der Signaturwert
1858 eingesetzt werden.

1859 **5.3.3.14 X509CERTIFICATE**

1860 Anstelle der Variable **X509CERTIFICATE** MUSS im XML-Signaturlayout das Base64-kodierte
1861 Signaturzertifikat (X509-Zertifikat, kodiert gem. Abschnitt 4.4.4 in [4]) eingesetzt werden.

1862 **5.3.3.15 DSIGOBJECT**

1863 Wird eine Signatur-Methode angewandt, die eine enveloping XML-Signatur erwirkt, so MUSS
1864 das folgende Fragment anstelle der Variable **DSIGOBJECT** im XML-Signaturlayout eingesetzt
1865 werden:

```
1866 <dsig:Object Id="signed-data-object-SIGDATAOBJURI">  
1867 <dsig:Base64Content>BASE64CONTENT</dsig:Base64Content>  
1868 </dsig:Object>
```

1869 In diesem Fragment MUSS anstelle der Variable **SIGDATAOBJURI** der aus den
1870 Signaturparametern zusammengesetzte Wert `<ParamSigDataObjURI>` eingesetzt werden.

1871 In diesem Fragment MUSS anstelle der Variable **BASE64CONTENT** der Base64-kodierte, zu
1872 signierende Datenstrom (gem. angewandter Signaturmethode) eingesetzt werden.

1873 Wird eine Signatur-Methode angewandt, die eine detached XML-Signatur erwirkt, so ist kein
1874 XML-Signaturobjekt für die signierten Daten notwendig. Die Variable **DSIGOBJECT** im XML-
1875 Signaturlayout MUSS ersatzlos entfernt werden.

1876 **5.3.3.16 SIGNINGTIME**

1877 Anstelle der Variable **SIGNINGTIME** MUSS der Signaturzeitpunkt gem. [5] eingesetzt werden.

1878 **5.3.3.17 DIGESTVALUEX509CERTIFICATE**

1879 Anstelle der Variable **DIGESTVALUEX509CERTIFICATE** MUSS der Hash-Wert des
1880 Signaturzertifikates (lt. [5]) eingesetzt werden.

1881 **5.3.3.18 CERTIFICATEDIGESTMETHOD**

1882 Anstelle der Variable `CERTIFICATEDIGESTMETHOD` MUSS im XML-Signaturlayout die
1883 Hashmethode (URI) des Zertifikatdigests eingesetzt werden.

1884 **5.3.3.19 X509ISSUERNAME**

1885 Anstelle der Variable `X509ISSUERNAME` MUSS der Name des Ausstellers des
1886 Signaturzertifikates (lt. [5]) eingesetzt werden.

1887 **5.3.3.20 X509SERIALNUMBER**

1888 Anstelle der Variable `X509SERIALNUMBER` MUSS die Seriennummer des Signaturzertifikates
1889 (lt. [5]) eingesetzt werden.

1890 **5.3.3.21 MIMETYPE**

1891 Anstelle der Variable `MIMETYPE` MUSS der MIME-Type der zu signierenden Daten eingesetzt
1892 werden. Der MIME-Type ist von der angewandten Signaturmethode abhängig und MUSS im
1893 Zuge der Definition der Signaturmethode festgelegt werden.

1894 **5.4 Signaturparameter-Profil etsi-moc-1.0**

1895 **5.4.1 Charakteristik**

1896 *Parameter-Kennzeichnung:* etsi-moc-1.0

1897 *Signaturerstellungskomponente:* dieses Profil MUSS für die Open Source BKU "MOCCA"
1898 verwendet werden.

1899 *Einschränkungen bzgl. Signatur-* urn:pdfsigfilter:bka.gv.at:text:v1.1.0,
1900 *methoden* urn:pdfsigfilter:bka.gv.at:text:v1.2.0,
1901 urn:pdfsigfilter:bka.gv.at:binaer:v1.1.0

1902 *Verwendbarkeit:* EMPFOHLEN

1903 **5.4.2 Signaturparameter**

1904 Für dieses Signaturparameter-Profil wurden Signaturparameter definiert. Diese MÜSSEN nach
1905 der vorangestellten Kennzeichnung des Profils in die Signatur-Repräsentation eingefügt
1906 werden.

1907 Für die Formulierung der Signaturparameter MUSS die Definition gem. Abschnitt 2.2 angewandt
1908 werden. Es gilt bei diesem Signaturparameter-Profil jedoch die Maßgabe, dass das Feld
1909 `<SIGDEV_PROF>` den Wert `etsi-moc-1.0` haben MUSS.

1910 Der Signaturparameter `PARAM_L1` (siehe Abschnitt 2.2) stellt einen jeweils innerhalb einer
1911 Signatur konstanten Wert dar, der für die Bildung der 7 unten beschriebenen Einzelwerte
1912 herangezogen werden MUSS. Der Parameter `PARAM_L2` wird nicht verwendet und entfällt
1913 zusammen mit dem Delimiter-Zeichen "@".

1914 Aus dem Signaturparameter `PARAM_L1` MÜSSEN folgende Einzelwerte gebildet werden:

1915	<code><ParamSigId></code>	::= "Signature-" <code><PARAM_L1></code> "-1"
1916	<code><ParamSignedInfoId></code>	::= "SignedInfo-" <code><PARAM_L1></code> "-1"
1917	<code><ParamSigDataRefId></code>	::= "Reference-" <code><PARAM_L1></code> "-1"
1918	<code><ParamEtsiDataRefId></code>	::= "Reference-" <code><PARAM_L1></code> "-2"
1919	<code><ParamSigValueId></code>	::= "SignatureValue-" <code><PARAM_L1></code> "-1"
1920	<code><ParamEtsiSignedPropertiesId></code>	::= "SignedProperties-" <code><PARAM_L1></code> "-1"
1921	<code><ParamEtsiDataObjId></code>	::= "Object-" <code><PARAM_L1></code> "-1"

1922 Diese Werte werden mehrfach im Signaturlayout referenziert und verwendet.

1923 Die Verwendung dieser Einzelwerte wird in Abschnitt 5.4.3 festgelegt.

1924 Nachfolgend ein Beispiel zur Bildung der Einzelwerte auf Basis übergebener Signaturparameter
1925 (spezifisch für das vorliegende Signaturparameter-Profil):

1926 Signaturparameter lt. Signaturrepräsentation:

1927 etsi-moc-1.0@b2e01c95

1928 Daraus ergeben sich:

1929	<ParamSigId>	=	Signature-b2e01c95-1
1930	<ParamSignedInfoId>	=	SignedInfo-b2e01c95-1
1931	<ParamSigDataRefId>	=	Reference-b2e01c95-1
1932	<ParamEtsiDataRefId>	=	Reference-b2e01c95-2
1933	<ParamSigValueId>	=	SignatureValue-b2e01c95-1
1934	<ParamEtsiSignedPropertiesId>	=	SignedProperties-b2e01c95-1
1935	<ParamEtsiDataObjId>	=	Object-b2e01c95-1

1936 5.4.3 Signaturlayout

1937 Die Signaturen einer Bürgerkartenumgebung enthalten zeitabhängige Varianzen. Diese sind vor
1938 allem eine Reihe von XML-Attributen (ID-Attribute) die zur Referenzierung von XML-
1939 Elementen/-Knoten herangezogen werden. Diese variablen Attribute MÜSSEN in Form der
1940 Signaturparameter innerhalb der Signatur-Repräsentation verwaltet und bei der Rekonstruktion
1941 der Signatur entsprechend berücksichtigt werden.

1942 Eine XML-Signatur, die diesem Parameter-Profil entspricht, MUSS dem folgenden Layout
1943 entsprechen:

```
1944 <dsig:Signature xmlns:dsig="http://www.w3.org/2000/09/xmldsig#" Id="SIGID">
1945   <dsig:SignedInfo Id="SIGNEDINFOID">
1946     <dsig:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
1947     <dsig:SignatureMethod Algorithm="ALGORITHM" />
1948     <dsig:Reference Id="SIGDATAREFID" URI="urn:Document">
1949       <dsig:DigestMethod Algorithm="DATADIGESTMETHOD" />
1950       <dsig:DigestValue>DIGESTVALUESIGNEDDATA</dsig:DigestValue>
1951     </dsig:Reference>
1952     <dsig:Reference Id="ETSIDATAREFID"
1953       Type="http://uri.etsi.org/01903/v1.1.1#SignedProperties"
1954       URI="#xmlns(xades=http://uri.etsi.org/01903/v1.1.1%23)%20xpointer(id('ETSIDATAOBJID')/
1955       child::xades:QualifyingProperties/child::xades:SignedProperties)">
1956       <dsig:DigestMethod Algorithm="PROPERTIESDIGESTMETHOD" />
1957       <dsig:DigestValue>DIGESTVALUESIGNEDPROPERTIES</dsig:DigestValue>
1958     </dsig:Reference>
1959   </dsig:SignedInfo>
1960   <dsig:SignatureValue Id="SIGVALUEID">SIGNATUREVALUE</dsig:SignatureValue>
1961   <dsig:KeyInfo>
1962     <dsig:X509Data>
1963       <dsig:X509Certificate>X509CERTIFICATE</dsig:X509Certificate>
1964     </dsig:X509Data>
1965   </dsig:KeyInfo>
1966   <dsig:Object Id="ETSIDATAOBJID">
1967     <QualifyingProperties xmlns="http://uri.etsi.org/01903/v1.1.1#"
1968     xmlns:ns2="http://www.w3.org/2000/09/xmldsig#">
1969       <SignedProperties xmlns="http://uri.etsi.org/01903/v1.1.1#"
1970       xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
1971       xmlns:ns2="http://www.w3.org/2000/09/xmldsig#" Id="ETSISIGNEDPROPERTIESID">
1972         <SignedSignatureProperties>
1973         <SigningTime>SIGNINGTIME</SigningTime>
```

1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998

```
<SigningCertificate>  
  <Cert>  
    <CertDigest>  
      <DigestMethod Algorithm="CERTIFICATEDIGESTMETHOD"/>  
      <DigestValue>DIGESTVALUEX509CERTIFICATE</DigestValue>  
    </CertDigest>  
    <IssuerSerial>  
      <ns2:X509IssuerName>X509ISSUERNAME</ns2:X509IssuerName>  
      <ns2:X509SerialNumber>X509SERIALNUMBER</ns2:X509SerialNumber>  
    </IssuerSerial>  
  </Cert>  
</SigningCertificate>  
<SignaturePolicyIdentifier>  
  <SignaturePolicyImplied/>  
</SignaturePolicyIdentifier>  
</SignedSignatureProperties>  
<SignedDataObjectProperties>  
  <DataObjectFormat ObjectReference="#SIGDATAREFID">  
    <MimeType>MIMETYPE</MimeType>  
  </DataObjectFormat>  
</SignedDataObjectProperties>  
</SignedProperties>  
</QualifyingProperties>  
</dsig:Object>  
</dsig:Signature>
```

1999
2000

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

2001
2002

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

2003
2004
2005
2006

1. Kanonisierungsmethode: <http://www.w3.org/2001/10/xml-exc-c14n#>
2. Digest-Methode: Wird im Layout angegeben. Sonst mit dem Standardwert: <http://www.w3.org/2000/09/xmldsig#sha1>
3. Qualifying Properties: nach ETSI (<http://uri.etsi.org/01903/v1.1.1#>)

2007

5.4.3.1 SIGID

2008
2009

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

2010

5.4.3.2 SIGNEDINFOID

2011
2012

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

2013

5.4.3.3 ALGORITHM

2014
2015
2016

Legt den Signaturalgorithmus fest. Wenn dieser nicht im Signaturparameter definiert ist, werden folgende Standardwerte verwendet (in Abhängigkeit der Eigenschaften des Signatur-Schlüssels):

2017
2018

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

2019
2020

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

2021 **5.4.3.4 SIGDATAREFID**

2022 Anstelle der Variable **SIGDATAREFID** MUSS im XML-Signaturlayout der aus den
2023 Signaturparametern zusammengesetzte Wert `<ParamSigDataRefId>` eingesetzt werden.

2024 **5.4.3.5 DIGESTVALUESIGNEDDATA**

2025 Anstelle der Variable **DIGESTVALUESIGNEDDATA** MUSS im XML-Signaturlayout der Base64-
2026 kodierte Hash-Wert der ersten XML-Signaturreferenz eingesetzt werden.

2027 **5.4.3.6 DATADIGESTMETHOD**

2028 Anstelle der Variable **DATADIGESTMETHOD** MUSS im XML-Signaturlayout die Hashmethode
2029 (URI) der ersten XML-Signaturreferenz eingesetzt werden.

2030 **5.4.3.7 ETSIDATAREFID**

2031 Anstelle der Variable **ETSIDATAREFID** MUSS im XML-Signaturlayout der aus den
2032 Signaturparametern zusammengesetzte Wert `<ParamEtsiDataRefId>` eingesetzt werden.

2033 **5.4.3.8 ETSIDATAOBJID**

2034 Anstelle der Variable **ETSIDATAOBJID** MUSS im XML-Signaturlayout der aus den
2035 Signaturparametern zusammengesetzte Wert `<ParamEtsiDataObjId>` eingesetzt werden.

2036 **5.4.3.9 DIGESTVALUESIGNEDPROPERTIES**

2037 Anstelle der Variable **DIGESTVALUESIGNEDPROPERTIES** MUSS im XML-Signaturlayout der
2038 Base64-kodierte Hash-Wert der zweiten XML-Signaturreferenz eingesetzt werden.

2039 **5.4.3.10 PROPERTIESDIGESTMETHOD**

2040 Anstelle der Variable **PROPERTYDIGESTMETHOD** MUSS im XML-Signaturlayout die
2041 Hashmethode (URI) der zweiten XML-Signaturreferenz eingesetzt werden.

2042 **5.4.3.11 SIGVALUEID**

2043 Anstelle der Variable **SIGVALUEID** MUSS im XML-Signaturlayout der aus den
2044 Signaturparametern zusammengesetzte Wert `<ParamSigValueId>` eingesetzt werden.

2045 **5.4.3.12 SIGNATUREVALUE**

2046 Anstelle der Variable **SIGNATUREVALUE** MUSS im XML-Signaturlayout der Base64-kodierte
2047 Signaturwert eingesetzt werden.

2048 **5.4.3.13 X509CERTIFICATE**

2049 Anstelle der Variable **X509CERTIFICATE** MUSS im XML-Signaturlayout das Base64-kodierte
2050 Signaturzertifikat (X509-Zertifikat, kodiert gem. Abschnitt 4.4.4 in [4]) eingesetzt werden.

2051 **5.4.3.14 ETSISIGNEDPROPERTIESID**

2052 Anstelle der Variable **ETSI SIGNED PROPERTIES ID** MUSS im XML-Signaturlayout der aus den
2053 Signaturparametern zusammengesetzte Wert `<ParamEtsiSignedPropertiesId>`
2054 eingesetzt werden.

2055 **5.4.3.15 SIGNINGTIME**

2056 Anstelle der Variable **SIGNINGTIME** MUSS der Signaturzeitpunkt gem. [5] eingesetzt werden.

2057 **5.4.3.16 DIGESTVALUEX509CERTIFICATE**

2058 Anstelle der Variable **DIGESTVALUEX509CERTIFICATE** MUSS der Base64-kodierte Hash-
2059 Wert des Signaturzertifikates (lt. [5]) eingesetzt werden.

2060 **5.4.3.17 CERTIFICATEDIGESTMETHOD**

2061 Anstelle der Variable **CERTIFICATEDIGESTMETHOD** MUSS im XML-Signaturlayout die
2062 Hashmethode (URI) des Zertifikatdigests eingesetzt werden.

2063 **5.4.3.18 X509ISSUERNAME**

2064 Anstelle der Variable **X509ISSUERNAME** MUSS der Name des Ausstellers des
2065 Signaturzertifikates (lt. [5]) eingesetzt werden.

2066 **5.4.3.19 X509SERIALNUMBER**

2067 Anstelle der Variable **X509SERIALNUMBER** MUSS die Seriennummer des Signaturzertifikates
2068 (lt. [5]) eingesetzt werden.

2069 **5.4.3.20 MIMETYPE**

2070 Anstelle der Variable **MIMETYPE** MUSS der MIME-Type der zu signierenden Daten eingesetzt
2071 werden. Der MIME-Type ist von der angewandten Signaturmethode abhängig und MUSS im
2072 Zuge der Definition der Signaturmethode festgelegt werden.

2073

2074 **5.5 Signaturparameter-Profil etsi-bka-atrust-1.0**

2075 **5.5.1 Charakteristik**

2076 *Parameter-Kennzeichnung:* etsi-bka-atrust-1.0

2077 *Signaturerstellungskomponente:* dieses Profil MUSS für die a.trust Bürgerkartenumgebung
2078 angewendet werden.

2079 *Einschränkungen bzgl. Signatur-* dieses Profil MUSS mit Signaturmethoden verwendet
2080 *methoden* werden, die eine detached Signatur erzeugen; diese sind:
2081 urn:pdfsigfilter:bka.gv.at:text:v1.1.0,
2082 urn:pdfsigfilter:bka.gv.at:text:v1.2.0,
2083 urn:pdfsigfilter:bka.gv.at:binaer:v1.1.0

2084 *Anwendbarkeit:* EMPFOHLEN

2085 5.5.2 Signaturparameter

2086 Für dieses Signaturparameter-Profil KÖNNEN Signaturparameter zur Spezifikation von
2087 Signatur-Suite und Hashalgorithmen (SIGDEV_SPEC) verwendet werden um von den
2088 Standardeinstellungen abweichende Werte zu repräsentieren.

2089 5.5.3 Signaturlayout

2090 Eine XML-Signatur, die diesem Parameter-Profil entspricht, MUSS dem folgenden Layout
2091 entsprechen und DARF NICHT variable (zeitabhängige oder zufällige) Werte enthalten:

```
2092 <dsig:Signature Id="signature-1-1" xmlns:dsig="http://www.w3.org/2000/09/xmldsig#">
2093   <dsig:SignedInfo xmlns:dsig="http://www.w3.org/2000/09/xmldsig#">
2094     <dsig:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-
2095     20010315"/>
2096     <dsig:SignatureMethod Algorithm="ALGORITHM"/>
2097     <dsig:Reference Id="reference-1-1" URI="urn:Document">
2098       <dsig:DigestMethod Algorithm="DATADIGESTMETHOD"/>
2099       <dsig:DigestValue>DIGESTVALUESIGNEDDATA</dsig:DigestValue>
2100     </dsig:Reference>
2101     <dsig:Reference Id="etsi-data-reference-1-1"
2102       Type="http://uri.etsi.org/01903/v1.1.1#SignedProperties" URI="">
2103       <dsig:Transforms>
2104         <dsig:Transform Algorithm="http://www.w3.org/2002/06/xmldsig-filter2">
2105           <xpf:XPath Filter="intersect" xmlns:etsi="http://uri.etsi.org/01903/v1.1.1#"
2106             xmlns:xpf="http://www.w3.org/2002/06/xmldsig-filter2"/>/**[@Id='etsi-signed-1-
2107 1']/etsi:QualifyingProperties/etsi:SignedProperties
2108           </xpf:XPath>
2109         </dsig:Transform>
2110       </dsig:Transforms>
2111       <dsig:DigestMethod Algorithm="PROPERTIESDIGESTMETHOD"/>
2112       <dsig:DigestValue>DIGESTVALUESIGNEDPROPERTIES</dsig:DigestValue>
2113     </dsig:Reference>
2114   </dsig:SignedInfo>
2115   <dsig:SignatureValue>SIGNATUREVALUE</dsig:SignatureValue>
2116   <dsig:KeyInfo>
2117     <dsig:X509Data>
2118       <dsig:X509Certificate>X509CERTIFICATE</dsig:X509Certificate>
2119     </dsig:X509Data>
2120   </dsig:KeyInfo>
2121   <dsig:Object Id="etsi-signed-1-1">
2122     <etsi:QualifyingProperties Target="#signature-1-1"
2123     xmlns:etsi="http://uri.etsi.org/01903/v1.1.1#">
2124       <etsi:SignedProperties xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
2125     xmlns:etsi="http://uri.etsi.org/01903/v1.1.1#">
2126         <etsi:SignedSignatureProperties>
2127           <etsi:SigningTime>SIGNINGTIME</etsi:SigningTime>
2128           <etsi:SigningCertificate>
2129             <etsi:Cert>
2130               <etsi:CertDigest>
2131                 <etsi:DigestMethod Algorithm="CERTIFICATEDIGESTMETHOD"/>
2132                 <etsi:DigestValue>DIGESTVALUEX509CERTIFICATE</etsi:DigestValue>
2133               </etsi:CertDigest>
2134               <etsi:IssuerSerial>
2135                 <dsig:X509IssuerName>X509ISSUERNAME</dsig:X509IssuerName>
2136                 <dsig:X509SerialNumber>X509SERIALNUMBER</dsig:X509SerialNumber>
2137               </etsi:IssuerSerial>
2138             </etsi:Cert>
2139           </etsi:SigningCertificate>
2140           <etsi:SignaturePolicyIdentifier>
2141             <etsi:SignaturePolicyImplied/>
2142           </etsi:SignaturePolicyIdentifier>
2143         </etsi:SignedSignatureProperties>
```

```
2144 <etsi:SignedDataObjectProperties>  
2145 <etsi:DataObjectFormat ObjectReference="#reference-1-1">  
2146 <etsi:MimeType>MIMETYPE</etsi:MimeType>  
2147 </etsi:DataObjectFormat>  
2148 </etsi:SignedDataObjectProperties>  
2149 </etsi:SignedProperties>  
2150 </etsi:QualifyingProperties>  
2151 </dsig:Object>  
2152 </dsig:Signature>
```

2153 Die vom Layout vorgesehenen Variabilitäten werden in Großbuchstaben und umrandet
2154 ausgewiesen. Die nachfolgenden Abschnitte definieren diese im Detail.

2155 Dieses Profil legt durch das damit festgelegte XML-Signaturlayout die folgenden wesentlichen
2156 XML-Signatur-Eigenschaften implizit fest:

2157 1. Kanonisierungsmethode:

2158 <http://www.w3.org/TR/2001/REC-xml-c14n-20010315>

2159 2. Digest-Methoden: Entweder im Layout angegeben. Sonst mit dem Standardwert:

2160 <http://www.w3.org/2000/09/xmlsig#sha1>

2161 3. Qualifying Properties: nach ETSI (<http://uri.etsi.org/01903/v1.1.1#>)

2162 5.5.3.1 ALGORITHM

2163 Legt den Signaturalgorithmus fest. Wenn dieser nicht im Signaturparameter definiert ist, werden
2164 folgende Standardwerte verwendet (in Abhängigkeit der Eigenschaften des Signatur-
2165 Schlüssels):

2166 • für ECDSA-Schlüssel: <http://www.w3.org/2000/09/xmlsig#rsa-sha1>

2167 • für RSA-Schlüssel: <http://www.w3.org/2000/09/xmlsig#rsa-sha1>

2168 Der zum Signaturschlüssel passende Identifikator (URI) MUSS im XML-Signaturlayout anstelle
2169 der Variable **ALGORITHM** eingesetzt werden.

2170 5.5.3.2 DIGESTVALUESIGNEDDATA

2171 Anstelle der Variable **DIGESTVALUESIGNEDDATA** MUSS im XML-Signaturlayout der Hash-
2172 Wert der ersten XML-Signaturreferenz eingesetzt werden.

2173 5.5.3.3 DATADIGESTMETHOD

2174 Anstelle der Variable **DATADIGESTMETHOD** MUSS im XML-Signaturlayout die Hashmethode
2175 (URI) der ersten XML-Signaturreferenz eingesetzt werden.

2176 5.5.3.4 DIGESTVALUESIGNEDPROPERTIES

2177 Anstelle der Variable **DIGESTVALUESIGNEDPROPERTIES** MUSS im XML-Signaturlayout der
2178 Hash-Wert der zweiten XML-Signaturreferenz eingesetzt werden.

2179 5.5.3.5 PROPERTIESDIGESTMETHOD

2180 Anstelle der Variable **PROPERTYDIGESTMETHOD** MUSS im XML-Signaturlayout die
2181 Hashmethode (URI) der zweiten XML-Signaturreferenz eingesetzt werden.

2182 5.5.3.6 SIGNATUREVALUE

2183 Anstelle der Variable **SIGNATUREVALUE** MUSS im XML-Signaturlayout der Signaturwert
2184 eingesetzt werden.

2185 **5.5.3.7 X509CERTIFICATE**

2186 Anstelle der Variable `X509CERTIFICATE` MUSS im XML-Signaturlayout das Base64-kodierte
2187 Signaturzertifikat (X509-Zertifikat, kodiert gem. Abschnitt 4.4.4 in [4]) eingesetzt werden.

2188 **5.5.3.8 SIGNINGTIME**

2189 Anstelle der Variable `SIGNINGTIME` MUSS der Signaturzeitpunkt gem. [5] eingesetzt werden.

2190 **5.5.3.9 DIGESTVALUEX509CERTIFICATE**

2191 Anstelle der Variable `DIGESTVALUEX509CERTIFICATE` MUSS der Hash-Wert des
2192 Signaturzertifikates (lt. [5]) eingesetzt werden.

2193 **5.5.3.10 CERTIFICATEDIGESTMETHOD**

2194 Anstelle der Variable `CERTIFICATEDIGESTMETHOD` MUSS im XML-Signaturlayout die
2195 Hashmethode (URI) des Zertifikatdigests eingesetzt werden.

2196 **5.5.3.11 X509ISSUERNAME**

2197 Anstelle der Variable `X509ISSUERNAME` MUSS der Name des Ausstellers des
2198 Signaturzertifikates (lt. [5]) eingesetzt werden.

2199 **5.5.3.12 X509SERIALNUMBER**

2200 Anstelle der Variable `X509SERIALNUMBER` MUSS die Seriennummer des Signaturzertifikates
2201 (lt. [5]) eingesetzt werden.

2202 **5.5.3.13 MIMETYPE**

2203 Anstelle der Variable `MIMETYPE` MUSS der MIME-Type der zu signierenden Daten eingesetzt
2204 werden. Der MIME-Type ist von der angewandten Signaturmethode abhängig und MUSS im
2205 Zuge der Definition der Signaturmethode festgelegt werden.

2206 **6 PDF Signature Field**

2207 Im Rahmen der Einbettung einer PDF-AS Signatur in ein PDF-Dokument KANN optional auch
2208 ein PDF-Spezifikation konformes Signaturfeld (siehe [2] Abschnitt 8.6.5, Signature Fields)
2209 erstellt werden. Damit wird das Dokument vor Signatur-brechenden Optimierungen geschützt,
2210 und die Existenz einer Signatur von jedem Reader erkannt.

2211 Das Signature Field wird mittels PDF-Dictionary repräsentiert und MUSS bzw. SOLL folgende
2212 Einträge enthalten (Details siehe [2] Table 8.60). Weitere Einträge KÖNNEN vorhanden sein.

Key	Value	Beschreibung
Type (MUSS)	Sig	Kennzeichnet das Signatur Dictionary
Filter (MUSS)	Adobe.PDF-AS	Beschreibt den Typ der Signatur und damit den SignaturHandler der zur Interpretation verwendet werden kann
SubFilter (MUSS)	z.B. urn:pdfsigfilter: bka.gv.at:text:v1.2.0	Beschreibt die verwendete Signaturmethode der PDF-AS Signatur.

Key	Value	Beschreibung
ContactInfo (SOLL)	z.B. https://www.buergerkarte.at/signature-verification	URL zum Prüfservice der PDF-AS Signatur
NAME (SOLL)	z.B. PDF-AS Signatur	Anzeigenname der Signatur
CONTENTS (MUSS)	<LEER>	Das Signatortoken MUSS leer sein. Die Daten zur Signatur sind je nach Signaturmethode laut PDF-AS Spezifikation im PDF enthalten.
BYTERANGE (MUSS)	Variabel, siehe [2] Table 8.60	Ein Array aus Integer-Paaren (Byte-Offset, Länge) welches den signierten Bereich kennzeichnet.

2213

2214 Eine optische Darstellung der Signatur KANN zusätzlich umgesetzt werden.

2215 7 BAIK Archiv Spezifikation

2216 Dieser Abschnitt beschreibt die Spezifikation von Signaturen für das BAIK-Archiv
2217 (Bundeskammer der Architekten und Ingenieurkonsulenten). Die Spezifikation ergibt sich aus
2218 Erweiterungen der Spezifikation zur PDF Amtssignatur PDF-AS 2.2.

2219 7.1 Darstellung des Signaturblocks

2220 7.1.1 Archivsignatur

2221 Für die Archivsignatur MUSS folgender Signaturblock verwendet werden.

ELEKTRONISCHE ARCHIVSIGNATUR		
Signaturwert	fsRvseYemgqIG+bfblUOfueYPAM4cr082xAtNXJJQUiORqPK22CQehMqTs5A9uYXdA2ycvhnwSyVApmyOpj1Z4W+cMyQkK7sxW4eDb8toB3XhtJmvqhu22/EKe8h3p7BGviiD2Q5+vE54eqSJjilQt3s5N7Bp/tqBi8S+pli2ULbJJ9vklJCg6NnbmE1wb1SNeTXaTluO2Vh9QQ8gGFdPMUYtSHMSiduK7POVj4+8bSivXNqnBuzoKkZ+Rne0D1c2HbNoarP2QOLzZiCjtblQfuStfgNf5g4rMnN9Nv1c3i6QeUzaG0i8vzJxFhA+FaaUKpInTDpavxqEBnCcqlh6A==	
	Signator	BAIK-Archiv Urkundenarchiv der Bundes-Architekten- und Ingenieurkonsulentenkammer
	Signaturdatum	2008-09-12T09:57:28+01:00
	Zertifizierungsdienst	CN=D-TRUST Qualified CA 3 2007:PN,O=D-Trust GmbH,C=DE
	Seriennummer	357875
	Algorithmus	http://www.w3.org/2001/04/xmldsig-more#rsa-sha256
	Methode	urn:pdfsigfilter:bka.gv.at:binaer:v1.1.0
Hinweis:	Dokumentenformat: PDF 1.4	

2222

Feld	Beschreibung
Überschrift	Der Text „ELEKTRONISCHE ARCHIVSIGNATUR“
Signaturwert	Signaturwert in BASE64 Codierung

Feld	Beschreibung
Signator	Fixe Texte BAIK-Archiv Urkundenarchiv der Bundeskammer für Architektur und Ingenieurekonsulenten
Signaturdatum	Datum und Uhrzeit der Signaturerstellung
Zertifizierungsdienst	Inhalt des Attributes „Aussteller“ des bei der Signatur verwendeten Zertifikats
Seriennummer	Inhalt des Attributes „Seriennummer“ des bei der Signatur verwendeten Zertifikats
Algorithmus	Identifizier des verwendeten Hashwert- und Signaturalgorithmus
Methode	Die URI für die Signaturmethode nach XXX
Hinweis	Der Text „Dokumentenformat“ + Bezeichner für das Format des signierten PDF Dokuments.

2223 Der Hintergrund des Signaturblockes ist in blauer Farbe dargestellt.

2224 7.1.2 Beurkundungssignatur

2225 Für die Beurkundungssignatur wird folgender Signaturblock verwendet.

ELEKTRONISCHE BEURKUNDUNGSSIGNATUR		
Signaturwert	FmPdABNDUwZoD2DSdfycDEF0TNUsYqa/lvtC0Kkaia4cwB38XvkMuP6Z7CzwSt2h	
staatlich befugter und beeideter  Ziviltechniker	Signator	Max Mustermann Ingenieurkonsulent für Vermessungswesen Kanzleisitz: Villach
	Signaturdatum	2009-09-07T11:22:33Z
	Zertifizierungsdienst	CN=a-sign-Premium-Sig-02,OU=a-sign-Premium-Sig-02, O=A-Trust Ges. f. Sicherheitssysteme im elektr. Da tenverkehr GmbH,C=AT
	Seriennummer	123456
	Algorithmus	http://www.w3.org/2007/05/xmldsig-more#ecdsa-ripemd160
	Methode	urn:pdfsigfilter:bka.gv.at:binaer:v1.0.0
Hinweis:	Dokumentenformat: ISO 19005-1:2005 PDF/A-1b	

2226

Feld	Beschreibung
Überschrift	Der Text „ELEKTRONISCHE BEURKUNDUNGSSIGNATUR“
Signaturwert	Signaturwert in BASE64 Codierung
Signator	Antragsteller -> T (optional) + Antragsteller -> G + Antragsteller -> SN Optional Antragsteller -> O + Antragsteller -> OU Text „Kanzleisitz“ + Antragsteller -> L

Signaturdatum	Datum und Uhrzeit der Signaturerstellung
Zertifizierungsdienst	Inhalt des Attributes „Aussteller“ des bei der Signatur verwendeten Zertifikats
Seriennummer	Inhalt des Attributes „Seriennummer“ des bei der Signatur verwendeten Zertifikats
Algorithmus	Identifizier des verwendeten Hashwert- und Signaturalgorithmus
Methode	Die URI für die Signaturmethode nach XXX
Hinweis	Der Text „Dokumentenformat“ + Bezeichner für das Format des signierten PDF Dokuments.

2227 Der Hintergrund des Signaturblockes ist in gelber Farbe dargestellt.

2228 **7.2 Erweiterungen zur PDF-AS Spezifikation**

2229 Dieser Abschnitt spezifiziert die Erweiterungen und Einschränkungen zur PDF-AS Spezifikation.

2230 **7.2.1 Neuer Platzhalter /alg**

2231 Der Platzhalter `/alg` MUSS den Algorithmus spezifizieren, welcher sowohl den Signatur- als
2232 auch den Hashwertalgorithmus definiert der bei der Signaturerstellung verwendet wird. In der
2233 Darstellung MUSS der URI der verwendeten Algorithmen-Suite verwendet werden.

2234 z.B.: <http://www.w3.org/2001/04/xmldsig-more#rsa-sha256>

2235 **7.2.2 Erweiterung des EGIZ Dictionary /TimeStamp**

2236 Das Element `/TimeStamp` enthält den elektronischen Zeitstempel, welcher optional durch das
2237 Archiv bei der Erstellung des Dokuments angebracht werden KANN.

2238 **Verwendung:** optional

2239 **Position:** unmittelbar nach `/Cert`

2240 **Aufbau:** Das Element ist äquivalent zum Element `/Cert` aufgebaut wobei jeder String statt
2241 einem X509 Zertifikat einen RFC 3161 Timestamp enthält.

2242 Der Zeitstempel KANN bei der Rückführung in die XMLDSig Struktur berücksichtigt werden.
2243 Beim XMLDSig ist der Zeitstempel nach der ETSI Xades Spezifikation einzubetten.

2244 **7.2.3 Verwendete Signaturmethode**

2245 Es MUSS die Signaturmethode `urn:pdfsigfilter:bka.gv.at:binaer:v1.1.0` mit den
2246 beschriebenen Erweiterungen verwendet werden. Es MUSS eine detached Signatur verwendet
2247 werden.

2248

2249

2250 **Referenzen**

- 2251 [1] ISO-19005-1:2005: Document management - Electronic document file format for long-
2252 term preservation - Part 1: Use of PDF 1.4 (PDF/A-1). ISO Standard 19005-1:2005.
- 2253 [2] Adobe Corporation: PDF Reference, third edition - Adobe portable document format
2254 version 1.4, ISBN 0-201-75839-3, December 2001, Addison-Wesley.
- 2255 [3] Adobe Corporation: PDF Reference, fifth edition - Adobe Portable Document Format
2256 version 1.6.
- 2257 [4] Eastlake, Donald, Reagle, Joseph und Solo, David: XML-Signature Syntax and
2258 Processing. W3C Recommendation, Februar 2002. Abgerufen aus dem World Wide
2259 Web am 14.05.2004 unter <http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/>.
- 2260 [5] ETSI XML Advanced Electronic Signatures (XAdES): ETSI TS 101 903 V1.2.2 (2004-
2261 04), Technical Specification.
- 2262 [6] Hollosi, Karlinger, et.al.: Die österreichische Bürgerkarte , Spezifikationsframework
2263 Version 1.2, März 2005. Abgerufen aus dem World Wide Web am 18.01.2008 unter
2264 <http://www.buergerkarte.at/konzept/securitylayer/spezifikation/aktuell/>.
- 2265 [7] Stabsstelle IKT-Strategie des Bundes: Spezifikation Module für Online Anwendungen –
2266 SP und SS, Version 1.3.0, vom 28. August 2005. Abgerufen aus dem World Wide Web
2267 am 10.01.2008 unter <http://egovlabs.gv.at/docman/view.php/6/20/MOA-SPSS-1.3.pdf>.
- 2268 [8] Rössler: Layout Amtssignatur – Spezifikation. Version 1.0.0 vom 25.06.2007.
- 2269 [9] RFC 2234: Augmented BNF for Syntax Specifications: ABNF