



<b>XML Definition der Personenbindung</b> 14. 02. 2005	<b>Konvention</b>
	Personenbindung - 1.2.2
	<b>Öffentlicher Entwurf</b>

Bezeichnung	XML-Definition der Personenbindung
Kurzbezeichnung	Personenbindung
Version	1.2.2
Datum	14. 02. 2005
Dokumentenklasse	Konvention
Dokumentenstadium	Öffentlicher Entwurf
Kurzbeschreibung	<p>Die Personenbindung ist integraler Bestandteil des Konzepts Bürgerkarte. Sie ist eine von der Behörde signierte Struktur, welche ein eindeutiges Identifikationsmerkmal einer Person (zum Beispiel eine Registernummer) einem oder mehreren Zertifikaten dieser Person zuordnet.</p> <p>Als solches dient die Personenbindung der eindeutigen, automatisierbaren Identifikation einer Person, wenn sie im Zuge eines Verfahrens an die Behörde herantritt.</p> <p>Dieses Papier beschreibt die XML-Spezifikation der Personenbindung.</p>
Autoren	Arno Hollosi, <a href="mailto:arno.hollosi@cio.gv.at">arno.hollosi@cio.gv.at</a> Gregor Karlinger, <a href="mailto:gregor.karlinger@cio.gv.at">gregor.karlinger@cio.gv.at</a>
Arbeitsgruppe	Stabsstelle IKT-Strategie des Bundes, Technik und Standards

# Inhalt

1	<b>Inhalt</b>	
2	Inhalt.....	2
3	1 Einleitung und Basisdaten .....	3
4	2 XML-Grundstruktur .....	4
5	2.1 SAML Assertion (Rahmenstruktur) .....	4
6	2.1.1 Beispiel .....	5
7	2.2 SAML Attribute Statement.....	5
8	2.2.1 Personendaten.....	5
9	2.2.2 Attribute.....	7
10	2.2.3 Beispiel .....	7
11	2.3 Die elektronische Signatur.....	8
12	3 Kodierungsvorschriften .....	10
13	4 Komprimierte Darstellung.....	11
14	4.1 ASN.1 Spezifikation.....	11
15	4.1.1 Erklärung zu einzelnen Feldern.....	12
16	5 Beispiel .....	13
17	5.1 Beispiel einer Personenbindung .....	13
18	5.2 Beispiel für komprimierte Darstellung.....	14
19	Referenzen.....	16
20	Historie .....	17

21 Dieses Dokument verwendet die Schlüsselwörter MUSS, DARF NICHT, ERFORDERLICH, SOLLTE,  
22 SOLLTE NICHT, EMPFOHLEN, DARF, und OPTIONAL zur Kategorisierung der Anforderungen. Diese  
23 Schlüsselwörter sind analog zu ihren englischsprachigen Entsprechungen MUST, MUST NOT,  
24 REQUIRED, SHOULD, SHOULD NOT, RECOMMENDED, MAY, und OPTIONAL zu handhaben, deren  
25 Interpretation in RFC 2119 festgelegt ist.

## 26 **1 Einleitung und Basisdaten**

27 Im Zuge von elektronischen Verfahren ist es für die Behörde wichtig, eine Person eindeutig zu  
28 identifizieren. Zertifikate wie sie für die elektronische Signatur verwendet werden, reichen für  
29 eine automatisierte, eindeutige Identifikation nicht aus, da sie meist nur den Namen der Person  
30 enthalten. Der Name einer Person ist für eine eindeutige Identifikation aber nicht  
31 ausreichend.

32 Aus diesem Grund wird die Person über ihre Stammzahl identifiziert, die für die Lebensdauer  
33 der Person konstant ist.

34 Die Personenbindung enthält neben der Stammzahl (dem Ordnungsbegriff für die Person) auch  
35 noch einen eindeutigen Bezeichner für jedes Zertifikat, dem die Stammzahl zugeordnet wird.  
36 Damit ist eine kryptographisch gesicherte Bindung zwischen der elektronischen Unterschrift  
37 einer Person (dem Signator) und eines für diese Person eindeutigen Identifikationsmerkmals  
38 gegeben.

## 39 2 XML-Grundstruktur

40 Die XML-Grundstruktur basiert auf der Security Assertion Markup Language [SAML 1.0]  
41 definiert von OASIS (Organization for the Advancement of Structured Information Standards).  
42 [SAML 1.0] definiert XML-Strukturen, die die Bestätigung (Assertions) von bestimmten  
43 Sachverhalten bzw. Beziehungen zwischen Subjekten durch Dritte (so genannte Authorities)  
44 zum Inhalt haben.

45 Im Falle der Personenbindung bestätigt dabei die Stammzahlenregisterbehörde die Beziehung  
46 der Stammzahl zu einem oder mehreren Zertifikaten.

47 Die Stammzahlenregisterbehörde sichert dabei durch ihre Signatur diese Beziehung  
48 kryptographisch gegen Veränderung ab. Die Signatur garantiert also die Authentizität der Daten  
49 und identifiziert die ausstellende Behörde über ihr Zertifikat.

50 Für die Personenbindung kommen folgende Standards und Spezifikationen zum Einsatz:

- 51 • Security Assertion Markup Language (SAML) – OASIS : Rahmenstruktur  
52 Namespace: `urn:oasis:names:tc:SAML:1.0:assertion`, Präfix: `saml`
- 53 • XML Digital Signatures (XMLDSIG) – W3C : elektronische Signaturen  
54 Namespace: `http://www.w3.org/2000/09/xmldsig#`, Präfix: `dsig`
- 55 • PersonData – CIO Austria : Platzhalter für Personendaten  
56 Namespace: `http://reference.e-government.gv.at/namespaces/  
57 persondata/20020228#`, Präfix: `pr`
- 58 • Vorschlag zur komprimierten Personenbindung – CIO Austria : Schema zur komprimierten  
59 Speicherung der Personenbindung  
60 Namespace: `http://www.buergerkarte.at/namespaces/  
61 personenbindung/20020506#`, Präfix: `il`

### 62 2.1 SAML Assertion (Rahmenstruktur)

63 Basis der Personenbindung ist die `saml:Assertion` Struktur aus [SAML 1.0].

64 Folgende verpflichtende Attribute sind im `saml:Assertion` Element enthalten:

Name	Wert	Beschreibung
MajorVersion	1	SAML Versionsnummer
MinorVersion	0	SAML Versionsnummer
AssertionID	xs:string	ID für die Assertion
Issuer	xs:string	Name des Ausstellers der Assertion.
IssueInstant	xs:dateTime	Zeitpunkt der Ausstellung der Personenbindung

65 Die `AssertionID` SOLLTE über die Applikationsgrenze hinweg eindeutig sein. Es wird  
66 EMPFOHLEN den Domainnamen der ausstellenden Behörde plus eine laufende Nummer bzw.  
67 die aktuelle Zeit zu verwenden (z.B. `bka.gv.at+2004-02-24T12:00:00.000Z`).

68 `Issuer` bezeichnet den Aussteller der `saml:Assertion` und MUSS im Kontext der  
69 Personenbindung ein URL sein, welcher auf eine Ressource verweist, die Namen, Anschrift und  
70 Signaturzertifikat des Ausstellers sowie optional weitere Informationen beinhaltet.  
71 Üblicherweise wird diese Information auf einer öffentlich zugänglichen Webseite

72 zusammengefasst werden. Der angegebene URL SOLLTE über einen großen Zeitraum konstant  
 73 sein, da er in verschiedenen Programmen als Parameter inkludiert sein kann.

74 Weiters sind im Kontext der Personenbindung genau folgende Elemente in der  
 75 saml:Assertion Struktur verpflichtend einzubinden:

Name	Beschreibung
saml:AttributeStatement	Enthält die Kerndaten der Personenbindung
dsig:Signature	Die elektronische Signatur des Ausstellers der Bindung.

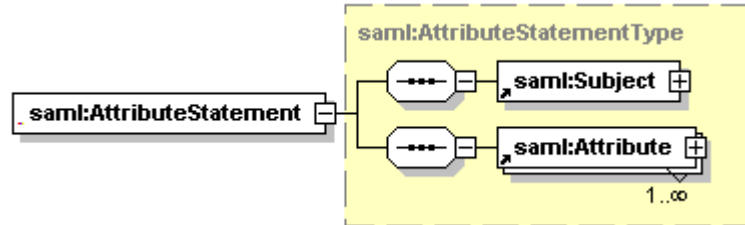
### 2.1.1 Beispiel

```

76
77 <?xml version="1.0" encoding="UTF-8"?>
78 <saml:Assertion
79   AssertionID="bka.gv.at+2004-02-24T12:00:00.000Z"
80   IssueInstant="2004-02-24T12:00:00.000Z"
81   Issuer="http://www.bka.gv.at/datenschutz/Stammzahlenregisterbehoerde"
82   MajorVersion="1"
83   MinorVersion="0"
84   <saml:AttributeStatement>
85     ...
86   </saml:AttributeStatement>
87   <dsig:Signature>
88     ...
89   </dsig:Signature>
90 </saml:Assertion>
  
```

## 2.2 SAML Attribute Statement

92 Das eingebundene saml:AttributeStatement enthält die Kerndaten der Personenbindung:



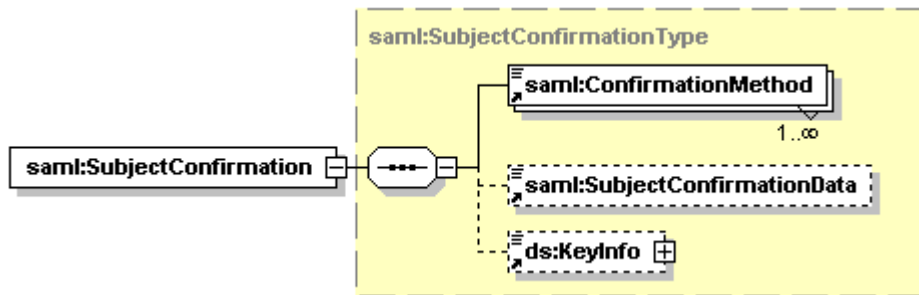
Name	Beschreibung
saml:Subject	Personendaten
saml:Attribute	Bezeichner für ein Zertifikat, dem die Stammzahl zugeordnet werden soll (verwendet wird als Bezeichner der öffentliche Schlüssel aus dem Zertifikat).

94 Dabei enthält saml:Subject die Daten der Person in Form der Personendaten-Struktur [PDat]  
 95 und die saml:Attribute Elemente jeweils einen öffentlichen Schlüssel als Bezeichner eines  
 96 zuzuordnenden Zertifikats in Form eines der dsig:KeyValue Sub-Elemente.

### 2.2.1 Personendaten

98 Die Struktur saml:Subject enthält genau das Element saml:SubjectConfirmation. In  
 99 diesem Element wird saml:ConfirmationMethod auf den Wert

100 urn:oasis:names:tc:SAML:1.0:cm:sender-vouches<sup>1</sup> gesetzt, während  
 101 saml:SubjectConfirmationData enthält die Daten der Person in Form eines pr:Person  
 102 Elements.



103

### 104 2.2.1.1 Personendaten für natürliche Personen

105 Das Element `pr:Person` MUSS vom Typ `pr:PhysicalPersonType` sein und genau  
 106 folgende Daten enthalten:

Name	Beschreibung
<code>pr:Identification</code>	Die Stammzahl der Person. Enthält genau ein Element <code>pr:Type</code> mit Inhalt <code>urn:publicid:gv.at:baseid</code> , und ein Element <code>pr:Value</code> , das die base64 kodierte Stammzahl als Stringwert enthält.
<code>pr:Name</code>	Der Name der natürlichen Person. Enthält genau ein Element <code>pr:GivenName</code> (Vorname) und ein Element <code>pr:FamilyName</code> mit Attribut <code>primary="undefined"</code> (Familiennamen). Mehrere Vornamen bzw. Mehrfach-Familiennamen MÜSSEN in einem Element zusammengefasst werden.
<code>pr:DateOfBirth</code>	Geburtsdatum der Person

### 107 Beispiel

```

108 <pr:Person
109   xsi:type="pr:PhysicalPersonType"
110   xmlns:pr="http://reference.e-government.gv.at/namespace/persondata/20020228#">
111   <pr:Identification>
112     <pr:Value>MDEyMzQ1Njc4OWFiY2RlZg==</pr:Value>
113     <pr:Type>urn:publicid:gv.at:baseid</pr:Type>
114   </pr:Identification>
115   <pr:Name>
116     <pr:GivenName>Herbert</pr:GivenName>
117     <pr:FamilyName primary="undefined">Gramgebeugt</pr:FamilyName>
118   </pr:Name>
119   <pr:DateOfBirth>1950-12-31</pr:DateOfBirth>
120 </pr:Person>

```

### 121 2.2.1.2 Personendaten für nichtnatürliche Personen

122 Personendaten für nichtnatürliche Personen (Firmen, Vereine, ...) werden in einer späteren  
 123 Version dieses Dokumentes definiert.

<sup>1</sup> Der Sender (die Behörde) bürgt für den Inhalt. Der Empfänger kann den Wahrheitsgehalt der Kerndaten nicht überprüfen (nicht zu verwechseln mit der Prüfung der Authentizität der Daten mittels der Signatur der Behörde).

124

## 2.2.2 Attribute

125

### 2.2.2.1 Attribute für natürliche Personen

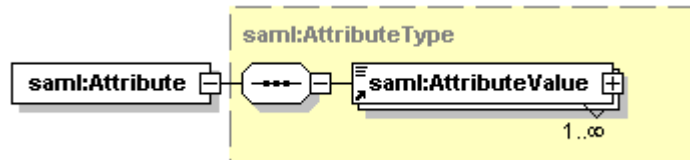
126

Die Struktur `saml:Attribute` MUSS mindestens einmal, KANN auch mehrere Male vorkommen und enthält je einen öffentlichen Schlüssel der Person in Form von `dsig:KeyValue` Sub-Elementen, also `dsig:RSAKeyValue`, `dsig:DSAKeyValue` oder `xsd:any` (zur Speicherung von ECDSA-Schlüsselwerten).

127

128

129



130

131

Im Attribut `AttributeName` ist der fixe Wert `CitizenPublicKey` anzugeben, im Attribut `AttributeNameSpace` der fixe Wert `urn:publicid:gv.at:namespaces:identitylink:1.2`.

132

133

134

### 2.2.2.2 Attribute für nichtnatürliche Personen

135

Attribute für nichtnatürliche Personen (Firmen, Vereine, ...) sind noch zu definieren.

136

## 2.2.3 Beispiel

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

159

160

161

162

163

164

165

166

167

168

169

170

171

172

173

174

175

176

177

178

179

```

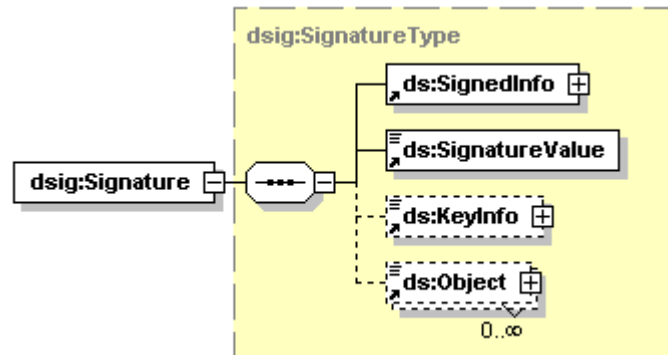
<saml:AttributeStatement
  xmlns:pr="http://reference.e-government.gv.at/namespace/persondata/20020228#"
  xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <saml:Subject>
    <saml:SubjectConfirmation>
      <saml:ConfirmationMethod>
        urn:oasis:names:tc:SAML:1.0:cm:sender-vouches</saml:ConfirmationMethod>
      <saml:SubjectConfirmationData>
        <pr:Person xsi:type="pr:PhysicalPersonType">
          <pr:Identification>
            <pr:Value>MDEyMzQ1Njc4OWFiY2RlZg==</pr:Value>
            <pr:Type>urn:publicid:gv.at:baseid</pr:Type>
          </pr:Identification>
          <pr:Name>
            <pr:GivenName>Herbert</pr:GivenName>
            <pr:FamilyName primary="undefined">Gramgebeugt</pr:FamilyName>
          </pr:Name>
          <pr:DateOfBirth>1950-12-31</pr:DateOfBirth>
        </pr:Person>
      </saml:SubjectConfirmationData>
    </saml:SubjectConfirmation>
  </saml:Subject>
  <saml:Attribute
    AttributeName="CitizenPublicKey"
    AttributeNamespace="urn:publicid:gv.at:namespaces:identitylink:1.2">
    <saml:AttributeValue>
      <dsig:RSAKeyValue xmlns:dsig="http://www.w3.org/2000/09/xmldsig#">
        <dsig:Modulus>...</dsig:Modulus>
        <dsig:Exponent>dG+9</dsig:Exponent>
      </dsig:RSAKeyValue>
    </saml:AttributeValue>
  </saml:Attribute>
  <saml:Attribute
    AttributeName="CitizenPublicKey"
    AttributeNamespace="urn:publicid:gv.at:namespaces:identitylink:1.2">
    <saml:AttributeValue>
      <dsig:RSAKeyValue xmlns:dsig="http://www.w3.org/2000/09/xmldsig#">
        <dsig:Modulus>...</dsig:Modulus>
        <dsig:Exponent>Q9Hf8w1UM3mKwROWcuWiz6Aucq8=</dsig:Exponent>
      </dsig:RSAKeyValue>
    </saml:AttributeValue>
  </saml:Attribute>

```

180 </saml:AttributeStatement>

## 181 2.3 Die elektronische Signatur

182 Die elektronische Signatur der saml:Assertion ist stark an das in [SAML 1.1] spezifizierte  
183 Profil von [XMLDSig] angelehnt.



184

185 Die Signatur enthält zwei dsig:Reference Elemente, die wie folgt ausgeführt werden  
186 MÜSSEN:

```
187 <dsig:Signature xmlns:dsig="http://www.w3.org/2000/09/xmldsig#">
188   <dsig:SignedInfo>
189     ...
190     <dsig:Reference
191       URI="#bka.gv.at+2004-02-24T12:00:00.000Z">
192       <dsig:Transforms>
193         <dsig:Transform Algorithm="http://www.w3.org/TR/1999/REC-xpath-19991116">
194           <dsig:XPath>not(ancestor-or-self::pr:Identification)</dsig:XPath>
195         </dsig:Transform>
196         <dsig:Transform
197           Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
198         <dsig:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
199         </dsig:Transforms>
200         <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
201         <dsig:DigestValue>GwNaF71Mc3mnpua+DJxwN8BG9Ww=</dsig:DigestValue>
202       </dsig:Reference>
203       <dsig:Reference
204         Type="http://www.w3.org/2000/09/xmldsig#Manifest"
205         URI="#bka.gv.at+2004-02-24T12:00:00.000Z">
206         <dsig:Transforms>
207           <dsig:Transform Algorithm="http://www.w3.org/TR/1999/REC-xpath-19991116">
208             <dsig:XPath>ancestor-or-self::dsig:Manifest</dsig:XPath>
209           </dsig:Transform>
210           <dsig:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
211           </dsig:Transforms>
212           <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
213           <dsig:DigestValue>1JdggeCTzaZ/TAgbOpxoc46+eEY=</dsig:DigestValue>
214         </dsig:Reference>
215       </dsig:SignedInfo>
216     ...
217   </dsig:Signature>
```

188 Die Referenzen DÜRFEN auch mit anderen mit Mitteln und Transformationen ausgeführt werden,  
189 solange das Ergebnis identisch mit dem der aus den hier angeführten Referenzen resultierenden  
200 Ergebnis ist. Es wird jedoch EMPFOHLEN, den in den Beispielen gezeigten  
201 Referenzierungsmechanismus zu verwenden, der aus dem Profil für [XMLDSig] in [SAML 1.1]  
202 stammt.  
222



223 Das zugehörige Manifest MUSS wie folgt aussehen:

```
224 <dsig:Signature xmlns:dsig="http://www.w3.org/2000/09/xmldsig#">
225   <dsig:SignedInfo>
226     ...
227   </dsig:SignedInfo>
228   ...
229   <dsig:Object>
230     <dsig:Manifest>
231       <dsig:Reference
232         URI="#bka.gv.at+2004-02-24T12:00:00.000Z">
233         <dsig:Transforms>
234           <dsig:Transform
235             Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
236           <dsig:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
237         </dsig:Transforms>
238         <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
239         <dsig:DigestValue>JaKFnaY5X742Xwk6KWz1Q5fa034=</dsig:DigestValue>
240       </dsig:Reference>
241     </dsig:Manifest>
242   </dsig:Object>
243 </dsig:Signature>
```

244 Auch hier gilt wieder, dass die Referenz mit anderen Mitteln und Transformationen ausgeführt  
245 werden DARF, solange das Ergebnis identisch mit dem der aus den hier angeführten Referenzen  
246 resultierenden Ergebnis ist.

247 Das erste dsig:Reference Element referenziert die ganze saml:Assertion mit Ausnahme  
248 der Stammzahl. Das Attribut URI weist dabei auf das Dokument-Element, die  
249 saml:Assertion. Es werden der Reihe nach eine XPath-Transformation [XPath] welche die  
250 Stammzahl ausnimmt und die Enveloped-Signature Transformation durchgeführt.

251 Die zweite Referenz bezieht sich auf das Manifest. Das Manifest selbst enthält eine einzelne  
252 Referenz, die auf die vollständige saml:Assertion verweist. Die Transformation nimmt  
253 wieder das dsig:Signature Element aus.

254 Mit diesem Aufbau der Signatur ist es möglich, die Stammzahl aus der Personenbindung zu  
255 entfernen und trotzdem eine validierende Signatur nach XMLDSig zu haben.

256 Bei der erweiterten Validierung (Validierung des Manifests) ist die Stammzahl jedoch mit  
257 eingeschlossen.

258 Weiters enthält die Signatur ein ds:KeyInfo Element, welches ausreichend Information für  
259 eine automatische Validierung der Signatur enthalten muss. Jedenfalls MUSS das X509  
260 Signaturzertifikat eingebunden werden.

261 Ein vollständiges Beispiel einer Personenbindung ist in Abschnitt 5 zu finden.

262

## 3 Kodierungsvorschriften

263

Um die im folgenden Kapitel beschriebene komprimierte Darstellung nutzen zu können, sind folgende Kodierungsvorschriften verpflichtend umzusetzen:

264

265

- Für Base-64 kodierte Werte in `saml:Attribute` sowie in `dsig:DigestValue` darf die Base64-Darstellung ausschließlich die Zeichen "a"-“z”, "A"-“Z”, "0"-“9”, "+" und "/", sowie abschließend (entsprechend den Base64-Regeln) bis zu zwei "=" verwenden. Zeilenumbrüche müssen nach exakt 76 Zeichen erfolgen; ist die abschließende Zeile 76 Zeichen lang wird vor dem End-Tag kein Zeilenumbruch mehr eingefügt.

266

267

268

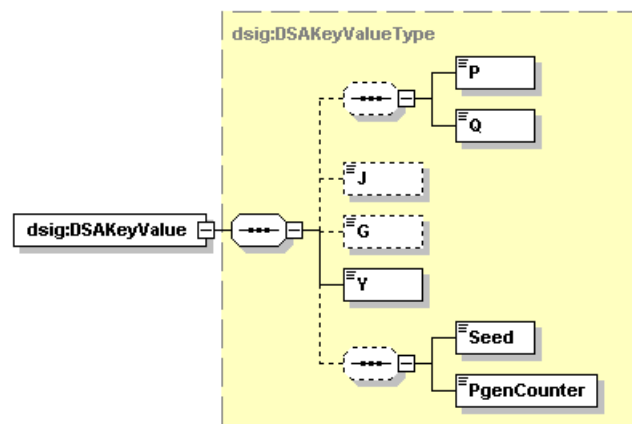
269

270

- Falls ein `dsig:DSAKeyValue` verwendet wird, dann sind genau die Parameter P, Q, G und Y anzugeben. Die Parameter J, seed und pGenCounter dürfen nicht angegeben werden. Siehe auch [RFC3279], Abschnitt 2.3.2.

271

272



## 273 4 Komprimierte Darstellung

274 Die nachfolgend beschriebene komprimierte Speicherung ist als Vorschlag zu verstehen und  
275 muss nicht verbindlich im Sinne dieser Spezifikation umgesetzt werden.

276 Die beschriebene XML-Struktur der Personenbindung hat eine Größe von ca. 5KB, was für  
277 Speicherung auf Smartcards problematisch sein kann, da diese nur eine sehr begrenzte  
278 Speichermöglichkeit haben. Ein großer Teil der Personenbindungsstruktur besteht jedoch aus  
279 bekannten und fixierten Werten, welche jederzeit nachgebildet werden können. Für die  
280 komprimierte Speicherung bietet es sich daher an, nur die variablen Teile zu speichern.

281 Der XML-Syntax erlaubt Variabilitäten innerhalb der definierten Struktur (Zeilenumbrüche, Ort  
282 der Namespace-Deklarationen, Kommentare, ...). Um die vorliegende Spezifikation allerdings  
283 nicht zu restriktiv zu gestalten wird folgendes Vorgehensmodell gewählt: die komprimierte  
284 Speicherung enthält eine URL auf einen XSLT-Stylesheet der Personenbindung. Als Protokolle  
285 sind HTTP und HTTPS zulässig. Der Stylesheet enthält dabei die komplette XML-Struktur der  
286 Personenbindung, der dann die variablen Teile hinzugefügt werden. Die URL des Stylesheets  
287 SOLLTE nicht länger als 48 Zeichen sein.

288 Damit wird einerseits dem Problem der Variabilität begegnet, andererseits ist die konkrete  
289 Ausprägung keinen in Zukunft vielleicht einengenden Bestimmungen unterworfen.

290 Die Kodierung der komprimierten Speicherung erfolgt als ASN.1 DER-kodierte Folge von  
291 Zeichen [ASN1] [DER]. Für die Umsetzung mittels Stylesheets ist das komprimierte Format  
292 zunächst in ein XML-File zu wandeln (XML-Typ `il:CompressedIdentityLink`). Die  
293 Elementnamen entsprechen dabei den Namen der Elemente im ASN.1<sup>3</sup>. Eine Applikation  
294 erzeugt ausgehend von den ASN.1 Daten den `il:CompressedIdentityLink`, lädt den  
295 XSLT-Stylesheet von der angegebenen URL und führt eine Transformation durch, um die  
296 ursprüngliche Personenbindung zu erhalten.

### 297 4.1 ASN.1 Spezifikation

```
298 PersonenBindung ::= SEQUENCE {  
299     version INTEGER,  
300     issuerTemplate UTF8String,  
301     assertionID UTF8String,  
302     issueInstant UTF8String,  
303     personData PersonData,  
304     citizenPublicKey SEQUENCE SIZE (1..MAX) OF CitizenPublicKey,  
305     signatureValue BIT STRING,  
306     referenceDigest [0] BIT STRING OPTIONAL,  
307     referenceManifestDigest [1] BIT STRING OPTIONAL,  
308     manifestReferenceDigest [2] BIT STRING OPTIONAL,  
309 }  
310  
311 PersonData ::= CHOICE {  
312     physcialPerson [0] PhysicalPersonData,  
313     corporateBody [1] CorporateBodyData  
314 }
```

---

<sup>3</sup> Abgesehen von der Groß- und Kleinschreibung.

```

315 PhysicalPersonData ::= SEQUENCE {
316     baseId UTF8String,
317     givenName UTF8String,
318     familyName UTF8String,
319     dateOfBirth UTF8String
320 }
321
322 CitizenPublicKey ::= CHOICE {
323     onToken [0] INTEGER,
324     referenceURL [1] UTF8String,
325     x509Data [2] SubjectPublicKeyInfo
326 }
327

```

328 Der Typ `CorporateBodyData` ist derzeit noch undefiniert.

### 329 4.1.1 Erklärung zu einzelnen Feldern

- 330 • *version*: bezeichnet die Version des Formats zur komprimierten Speicherung  
331 (wird nicht in XML Darstellung übernommen). In der vorliegenden ASN.1-Struktur ist  
332 das Feld auf den Wert „1“ zu setzen.
- 333 • *issuerTemplate*: ist die URL von der der Stylesheet geladen werden kann. Da davon  
334 ausgegangen werden kann, dass die Anzahl der Stylesheets sehr beschränkt ist, können  
335 Bürgerkarten-Umgebungen die Stylesheets auch cachen.
- 336 • *assertionID*: einzufüllen in das Attribut `AssertionID` von `saml:Assertion`. Dabei  
337 ist zu beachten, dass falls das Template in diesem Attribut bereits Zeichen enthält, die  
338 *assertionID* angehängt wird. Damit kann z.Bsp. der gleich bleibende Teil, der die Issuer  
339 voneinander unterscheidet, auch im Template aufgenommen werden.
- 340 • *personData* und dazugehöriger Typ *PhysicalPersonData*: sind in den entsprechenden  
341 Stellen einzusetzen.
- 342 • *citizenPublicKey*: bietet drei Möglichkeiten, einen öffentlichen Schlüssel als Bezeichner  
343 für das Zertifikat zu speichern:
  - 344 ○ *onToken*: die Information zur Gewinnung des öffentlichen Schlüssels befindet  
345 sich auf dem Security-Token (z.Bsp. in Form eines Zertifikats). Die Zahl dient  
346 als Ordnungsbegriff, falls mehrere solcher Informationen auf der Karte  
347 vorhanden sind – der Token gibt dabei die Nummerierung vor.
  - 348 ○ *referenceURL*: gibt eine URL an, unter der das bezeichnete DER-kodierte X509-  
349 Zertifikat abgerufen werden kann.
  - 350 ○ *x509Data*: enthält entsprechend der X509 Spezifikation [RFC2459] den  
351 öffentlichen Schlüssel (X509 Typ *SubjectPublicKeyInfo*)
  - 352 ○ Für die XML Darstellung ist *onToken* und *referenceURL* entsprechend  
353 aufzulösen und der Schlüsselwert einzusetzen.
- 354 • *signatureValue*: Wert der Signatur – muss noch Base64-kodiert werden, bevor er in die  
355 XML-Struktur eingefügt werden kann.
- 356 • *referenceDigest*, *referenceManifestDigest*, *manifestReferenceDigest*: optional die Hash-  
357 Werte der beiden Referenzen in der Signatur, sowie der Hash-Wert der Referenz im  
358 Manifest (*manifestReferenceDigest*). Die Werte müssen ebenfalls noch Base64-kodiert  
359 werden, bevor sie in die XML-Struktur eingefügt werden können. Optional deshalb, weil  
360 beim Befüllen der XML-Struktur diese Werte errechnet werden können. In der  
361 komprimierten XML-Struktur (Eingangsdaten für den Stylesheet) sind diese Felder aber

362 verpflichtet anzuführen, damit der Stylesheet die entsprechenden Felder in der  
363 Personenbindung ausfüllen kann.

364 Mit dieser Struktur kann eine Personenbindung in 100-150 Bytes gespeichert werden, wenn sich  
365 die öffentlichen Schlüssel abrufbar auf dem Token befinden, bzw. in ca. 400 Byte gespeichert  
366 werden, wenn die öffentlichen Schlüssel inkludiert werden.

367 Bei der Umwandlung der Personenbindung in die komprimierte ASN.1 Darstellung bzw. in die  
368 komprimierte XML-Darstellung (il:CompressedIdentityLink) muss darauf geachtet  
369 werden, die Werte identisch zu übernehmen, im Speziellen sind bei Elementwerten führende  
370 oder abschließende Leerzeichen oder Zeilenumbrüche mit zu übernehmen, um eine Bit-ident  
371 Rekonstruktion zu ermöglichen.

## 372 5 Beispiel

373 Das Beispiel zeigt eine vollständige Personenbindung für eine natürliche Person, mit Ausnahme  
374 der Werte der kryptographischen Daten (Hash-, Signatur, Schlüssel- und Zertifikatswerte).

### 375 5.1 Beispiel einer Personenbindung

```
376 <?xml version="1.0" encoding="UTF-8"?>
377 <saml:Assertion
378   AssertionID="bka.gv.at+2004-02-24T12:00:00.000Z"
379   IssueInstant="2004-02-24T12:00:00.000Z"
380   Issuer="http://www.bka.gv.at/datenschutz/Stammzahlenregisterbehoerde"
381   MajorVersion="1"
382   MinorVersion="0"
383   xmlns:pr="http://reference.e-government.gv.at/namespace/persondata/20020228#"
384   xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"
385   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
386   <saml:AttributeStatement>
387     <saml:Subject>
388       <saml:SubjectConfirmation>
389         <saml:ConfirmationMethod>
390           urn:oasis:names:tc:SAML:1.0:cm:sender-vouches</saml:ConfirmationMethod>
391         <saml:SubjectConfirmationData>
392           <pr:Person xsi:type="pr:PhysicalPersonType">
393             <pr:Identification>
394               <pr:Value>MDEyMzQ1Njc4OWFiY2RlZg==</pr:Value>
395               <pr:Type>urn:publicid:gv.at:baseid</pr:Type>
396             </pr:Identification>
397             <pr:Name>
398               <pr:GivenName>Herbert</pr:GivenName>
399               <pr:FamilyName primary="undefined">Gramgebeugt</pr:FamilyName>
400             </pr:Name>
401             <pr:DateOfBirth>1950-12-31</pr:DateOfBirth>
402           </pr:Person>
403         </saml:SubjectConfirmationData>
404       </saml:SubjectConfirmation>
405     </saml:Subject>
406     <saml:Attribute>
407       AttributeName="CitizenPublicKey"
408       AttributeNamespace="urn:publicid:gv.at:namespaces:identitylink:1.2">
409       <saml:AttributeValue>
410         <dsig:RSAKeyValue xmlns:dsig="http://www.w3.org/2000/09/xmldsig#">
411           <dsig:Modulus>...</dsig:Modulus>
412           <dsig:Exponent>...</dsig:Exponent>
413         </dsig:RSAKeyValue>
414       </saml:AttributeValue>
415     </saml:Attribute>
416     <saml:Attribute>
417       AttributeName="CitizenPublicKey"
418       AttributeNamespace="urn:publicid:gv.at:namespaces:identitylink:1.2">
419       <saml:AttributeValue>
420         <dsig:RSAKeyValue xmlns:dsig="http://www.w3.org/2000/09/xmldsig#">
```

```

421     <dsig:Modulus>...</dsig:Modulus>
422     <dsig:Exponent>...</dsig:Exponent>
423   </dsig:RSAKeyValue>
424 </saml:AttributeValue>
425 </saml:Attribute>
426 </saml:AttributeStatement>
427 <dsig:Signature xmlns:dsig="http://www.w3.org/2000/09/xmldsig#">
428   <dsig:SignedInfo>
429     <dsig:CanonicalizationMethod
430       Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
431     <dsig:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
432     <dsig:Reference URI="#bka.gv.at+2004-02-24T12:00:00.000Z">
433       <dsig:Transforms>
434         <dsig:Transform Algorithm="http://www.w3.org/TR/1999/REC-xpath-19991116">
435           <dsig:XPath>not(ancestor-or-self::pr:Identification)</dsig:XPath>
436         </dsig:Transform>
437         <dsig:Transform
438           Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
439         </dsig:Transforms>
440         <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
441         <dsig:DigestValue>Rv01PzN5sd4WVclcz/PTz/hqUIo=</dsig:DigestValue>
442       </dsig:Reference>
443     <dsig:Reference URI="#bka.gv.at+2004-02-24T12:00:00.000Z"
444       Type="http://www.w3.org/2000/09/xmldsig#Manifest">
445       <dsig:Transforms>
446         <dsig:Transform Algorithm="http://www.w3.org/TR/1999/REC-xpath-19991116">
447           <dsig:XPath>ancestor-or-self::dsig:Manifest</dsig:XPath>
448         </dsig:Transform>
449       </dsig:Transforms>
450       <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
451       <dsig:DigestValue>K/GKbymPaUtsr3Qh0De5uwHM9CU=</dsig:DigestValue>
452     </dsig:Reference>
453   </dsig:SignedInfo>
454   <dsig:SignatureValue>...</dsig:SignatureValue>
455   <dsig:KeyInfo>
456     <dsig:X509Data>
457       <dsig:X509Certificate>...</dsig:X509Certificate>
458       <dsig:X509Certificate>...</dsig:X509Certificate>
459       <dsig:X509Certificate>...</dsig:X509Certificate>
460     </dsig:X509Data>
461   </dsig:KeyInfo>
462   <dsig:Object>
463     <dsig:Manifest>
464       <dsig:Reference URI="#bka.gv.at+2004-02-24T12:00:00.000Z">
465         <dsig:Transforms>
466           <dsig:Transform
467             Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
468           </dsig:Transforms>
469           <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
470           <dsig:DigestValue>vHj9m+TpUI7zWjM+0QIgaiD/Lq0=</dsig:DigestValue>
471         </dsig:Reference>
472       </dsig:Manifest>
473     </dsig:Object>
474   </dsig:Signature>
475 </saml:Assertion>

```

## 5.2 Beispiel für komprimierte Darstellung

```

476 <CompressedIdentityLink
477   xmlns="http://www.buergerkarte.at/namespaces/personenbindung/20020506#"
478   xmlns:pr="http://reference.e-government.gv.at/namespace/persondata/20020228#"
479   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
480   xmlns:dsig="http://www.w3.org/2000/09/xmldsig#">
481   <IssuerTemplate>http://www.bka.gv.at/pathToStylesheet/Sheet.xsl</IssuerTemplate>
482   <AssertionID>bka.gv.at+2004-02-24T12:00:00.000Z</AssertionID>
483   <IssueInstant>2004-02-24T12:00:00.000Z</IssueInstant>
484   <PersonData xsi:type="pr:PhysicalPersonType">
485     <pr:Identification>
486       <pr:Value>MDEyMzQ1Njc4OWFiY2RlZg==</pr:Value>
487       <pr:Type/>
488     </pr:Identification>
489     <pr:Name>

```

```
491     <pr:GivenName>Herbert</pr:GivenName>
492     <pr:FamilyName>Grangebeugt</pr:FamilyName>
493 </pr:Name>
494 <pr:DateOfBirth>1950-12-31</pr:DateOfBirth>
495 </PersonData>
496 <CitizenPublicKey>
497   <dsig:RSAKeyValue>
498     <dsig:Modulus> .... </dsig:Modulus>
499     <dsig:Exponent> .... </dsig:Exponent>
500   </dsig:RSAKeyValue>
501 </CitizenPublicKey>
502 <CitizenPublicKey>
503   <dsig:DSAKeyValue>
504     <dsig:P> .... </dsig:P>
505     <dsig:Q> .... </dsig:Q>
506     <dsig:G> .... </dsig:G>
507     <dsig:Y> .... </dsig:Y>
508   </dsig:DSAKeyValue>
509 </CitizenPublicKey>
510 <SignatureValue> .... </SignatureValue>
511 <ReferenceDigest> .... </ReferenceDigest>
512 <ReferenceManifestDigest> .... </ReferenceManifestDigest>
513 <ManifestReferenceDigest> .... </ManifestReferenceDigest>
514 </CompressedIdentityLink>
```

## 515 Referenzen

### 516 ASN1

517 ITU-T Recommendation X.680 (1997), ISO/IEC 8824-1: 1998, Information Technology –  
518 Abstract Syntax Notation One (ASN.1), Specification of Basic Notation.

### 519 DER

520 ITU-T Recommendation X.690 (1997), ISO/IEC 8825-1: 1998, Information Technology –  
521 ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encodig  
522 Rules (CER) and Distinguished Encoding Rules (DER).

### 523 PersDat

524 Hollosi, Arno und Reichstädter, Peter: XML-Spezifikation der Personen-Daten Struktur.  
525 Konvention zum e-Government Austria erarbeitet vom CIO des Bundes, Operative Unit.  
526 Öffentlicher Entwurf, Version 1.0.1, 24. April 2002. Abgerufen aus dem World Wide  
527 Web am 24. Februar 2004unter <http://reference.e-government.at>.

### 528 RFC3279

529 W. Polk, R. Housley, L. Bassham: RFC 3279, Algorithms and Identifiers for the  
530 Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List  
531 (CRL) Profile, April 2002. Abgerufen aus dem World Wide Web am 24. Februar  
532 2004unter <http://www.ietf.org/rfc/rfc3279.txt>.

### 533 SAML 1.0

534 OASIS: Assertions and Protocol for the OASIS Security Assertion Markup Language  
535 (SAML). OASIS Standard, 5. November 2002.  
536 <http://www.oasis-open.org/committees/security/>

### 537 SAML 1.1

538 OASIS: Assertions and Protocol for the OASIS Security Assertion Markup Language  
539 (SAML) V1.1. OASIS Standard, 2. September 2003.  
540 <http://www.oasis-open.org/committees/security/>

### 541 XPath

542 James Clark, Steve DeRose: XML Path Language (XPath). W3C Recommendation,  
543 November 1999. Abgerufen aus dem World Wide Web am 24. Februar 2004unter  
544 <http://www.w3.org/TR/1999/REC-xpath-19991116>.

### 545 XMLDSig

546 Donald Eastlake, Joseph Reagle und David Solo: XML-Signature Syntax and Processing.  
547 W3C Recommendation, Februar 2002. Abgerufen aus dem World Wide Web am 24.  
548 Februar 2004unter <http://www.w3.org/TR/2002/REC-xmlsig-core-20020212>.



# Historie

Version	Datum	Kommentar
1.1.0	6. 5. 2002	
Ersteller		<ul style="list-style-type: none"> <li>• Umstellung auf SAML Version 1.0 vom 19. April 2002.</li> <li>• Vollständige Konformität mit SAML: <ul style="list-style-type: none"> <li>○ Verwendung von AttributeStatement statt eigenen Typs PersonAttributeStatement</li> <li>○ Assertion ohne ID Attribut – Signatur-Referenzen nun mit XPointer</li> </ul> </li> </ul> <p>Neue Referenzen: XPointer, XPath</p>
Arno Hollosi		
Version	Datum	Kommentar
1.1.1	6. 5. 2003	
Ersteller		<ul style="list-style-type: none"> <li>• Editoriale Verbesserungen</li> </ul>
Gregor Karlinger		
Version	Datum	Kommentar
1.2.0	14. 10. 2003	
Ersteller		<ul style="list-style-type: none"> <li>• Nomenklatur aktualisiert: Stammzahl, bereichsspezifische Personenkennung</li> <li>• Referenzen aktualisiert.</li> </ul> <p>Beispiele überarbeitet; Signatur an Profil aus [SAML 1.1] angenähert.</p>
Gregor Karlinger Arno Hollosi		
Version	Datum	Kommentar
1.2.1	24. 02. 2004	
Ersteller		<ul style="list-style-type: none"> <li>• Bezeichner für Aussteller der Personenbindung geändert.</li> <li>• Bezeichner für die Stammzahl geändert.</li> <li>• Namenraum für das SAML-Attribut CitizenPublicKey geändert.</li> </ul>
Gregor Karlinger		
Version	Datum	Kommentar
1.2.2	14. 02. 2005	
Ersteller		<ul style="list-style-type: none"> <li>• Statt der Formulierung „juristische Person“ wird nun die Formulierung „nichtnatürliche Person“ verwendet.</li> <li>• Zeile 247-250: Der erste Teil des zweiten Satzes lautet nun „Das Attribut URI weist dabei auf das Dokument-Element,“ statt wie bisher „Das Attribut URI weist dabei auf das Dokument,“.</li> <li>• Zeilen 247-250: Fußnote 2 entfernt, da widersinnig.</li> <li>• Zeile 411 lautet nunmehr korrekt: „&lt;dsig:Modulus&gt;...&lt;/dsig:Modulus&gt;“</li> </ul>
Gregor Karlinger		