Bundesministerium
für Umwelt, Naturschutz
und Reaktorsicherheit

Dokumentation

Definition der Schnittstelle für die Nachweisverordnung NachwV n.F.

Schnittstellenversion 1.04

Corrigendum 17.03.2009

Stand: 17.03.2009

Auftraggeber: Bundesministerium für Umwelt
Robert-Schumann-Platz 3
53175 Bonn

Dokument: Doc_BMU_05_203_V1_04_V03_Corrigendum_090317.doc
geprüft: 17.03.2009
freigegeben: 17.03.2009

Inhalt

1 Vorbemerkung

Dieses Dokument dient der Korrektur eines Fehlers in der Originaldokumentation der Version 1.04 der BMU-Schnittstelle. Alle hier nicht enthaltenen Teile der Originaldokumentation gelten unverändert weiter.

Die Korrektur umfasst nur die Variante 1 im Kapitel 4.4.2, die Variante 2 bleibt unverändert gegenüber der Ursprungsfassung dieser Dokumentation.

2 Allgemeines

3 Übergreifende Erläuterungen

4 Grundlegender Nachrichtenaufbau

4.1 Aufbau von <Nachricht>

4.2 Schemata-Organisation

4.3 Gemeinsame Bibliothek

4.4 Digitale Signaturen

4.4.1 Anforderungen an die Signaturdaten

4.4.2 Anforderungen an zweifache Signaturen

Die Signaturen können in den Dokumenten der BMU-Schnittstelle jeweils keinmal, einmal oder zweimal angebracht werden. Die zweifache Signatur eines Dokumentes dient der Unterstützung des ‚Vier-Augen-Prinzips', welches in vielen Unternehmen per Unterschriftenregelung verordnet ist.

Bei der zweiten Signatur kann entschieden werden, ob die beiden Signaturen unabhängig parallel nebeneinander stehen und jeweils nur das Dokument signieren, oder ob die zweite Signatur zusätzlich auch die erste Signatur mit signiert.

Die erste Signatur muss gegen die Änderungen durch eine nachfolgende zweite Signatur ‚geschützt' werden, damit sie ihre Gültigkeit nicht verliert. Wenn die Option einer Zweitsignatur eröffnet werden soll, muss also bereits die erste Signatur entsprechend formuliert sein, unabhängig davon, ob im Einzelfall tatsächlich eine zweite Signatur angebracht wird.

Nachfolgend zwei Varianten für XPath und XPath-Filter2:

Variante 1 (XPath in Verbindung mit <LayerID>-Attributen der Layer-Elemente):

```
<ds:Signature Id="beh-12345678-90AB-CDEF-1234-567890ABCDEF"
   xmlns:ds="http://www.w3.org/2000/09/xmldsig#">

      <ds:SignedInfo…>

            <ds:CanonicalizationMethod …/>
```

```
            <ds:SignatureMethod …/>

            <ds:Reference URI="" …>[1]

                  <ds:Transforms …>

                        <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-
                           signature"/>

                        <ds:Transform Algorithm="http://www.w3.org/TR/1999/REC-xpath-
                           19991116">

                              <ds:XPath>count(ancestor-or-
                                 self::en:ENSNBEHLayer[@lib:LayerID='BEH-4']) &gt;
                                 0</ds:XPath>

                        </ds:Transform>

                        <ds:Transform Algorithm="http://www.w3.org/TR/1999/REC-xpath-19991116"
                           xmlns:ds="http://www.w3.org/2000/09/xmldsig#">

                              <ds:XPath
                                 xmlns:ds="http://www.w3.org/2000/09/xmldsig#">count(ancestor-
                                 or-self::ds:Signature/preceding-
                                 sibling::ds:Signature/parent::node()[@lib:LayerID = 'BEH-4'])
                                 = 0</ds:XPath>

                        </ds:Transform>

                        <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>

                  </ds:Transforms>

                  <ds:DigestMethod …/>

                  <ds:DigestValue> …</ds:DigestValue>

            </ds:Reference>

      </ds:SignedInfo>

      <ds:SignatureValue …>…</ds:SignatureValue>

      <ds:KeyInfo …>

      …

      </ds:KeyInfo>

</ds:Signature>
```

Der XPath im zweiten <ds:Transform>-Container weist auf das <Id>-Attribut des Layers, zu dem die Signatur gehört (in diesem Beispiel handelt es sich also um ein Signatur des Layers mit der LayerID ‚BEH-4'). Der Ausdruck ‚`ancestor-or-self::en:ENSNBEHLayer[@lib:LayerID='BEH-4']`' ist doppelt restriktiv, da sowohl der genaue Names des Layers (`en:ENSNBEHLayer`) als auch die an sich ja schon eindeutige ID des Layers (`[@lib:LayerID='BEH-4']`) enthalten sind. Alternativ zu dieser exakten Angabe des Layerknotennamens kann auch

[1] Diese URI muss leer bleiben. In der ersten Veröffentlichung war hier ein Pointer auf die ID des Layers eingetragen worden. Da die ID des Layers aber nicht ‚Id', sondern ‚LayerID' heißt, funktioniert dieser Mechanismus nicht mit Standard-XML-Signature-Tools, die nach einem Attribut namens ‚Id' suchen. Angepasste Tools, welche die LayerID erkennen, verifizieren keine Standardsignaturen und Signaturen angepasster Tools können nicht von Original-Tools verifiziert werden.

mit //* (`ancestor-or-self:://*[@lib:LayerID='BEH-4']`) oder mit node() (`ancestor-or-self::node()[@lib:LayerID='BEH-4']`) operiert werden. Diese beiden Vereinfachungen wurden jedoch nicht im Rahmen dieser Dokumentation speziell getestet.

Der zweite XPath-Ausdruck prüft für jedes Element unterhalb des durch den vorhergehenden Ausdruck identifizierten Knotens, wie viele Signaturen mit einer weiteren vorherstehenden Signatur an einem Elternknoten mit eben jener Id ‚BEH-4' zu finden sind. Wenn die Anzahl 0 ist (dies gilt insbesondere auch für die erste Signatur des Layers) wird der Knoten mit signiert. Wenn die Anzahl nicht 0 ist (dies gilt insbesondere für die zweite Signatur), wird der Knoten samt Inhalt nicht mit signiert. Bei dieser Variante signiert die zweite Signatur also die erste Signatur mit!

Variante 2 (XPath-Filter2):

```
<ds:Signature Id="beh-12345678-90AB-CDEF-1234-567890ABCDEF" …>
      <ds:SignedInfo…>

            <ds:CanonicalizationMethod …/>

            <ds:SignatureMethod …/>

            <ds:Reference …>

                  <ds:Transforms …>

                        <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-
                           signature" xmlns:ds="http://www.w3.org/2000/09/xmldsig#"/>

                        <ds:Transform Algorithm="http://www.w3.org/2002/06/xmldsig-filter2"
                           xmlns:ds="http://www.w3.org/2000/09/xmldsig#">

                              <dsig-xpath:XPath Filter="intersect" xmlns:dsig-
                                 xpath="http://www.w3.org/2002/06/xmldsig-
                                 filter2">descendant::*[local-name()='ENSNBEHLayer'][last()-
                                 0]</dsig-xpath:XPath>

                        </ds:Transform>

                        <ds:Transform Algorithm="http://www.w3.org/2002/06/xmldsig-filter2"
                           xmlns:ds="http://www.w3.org/2000/09/xmldsig#">

                              <dsig-xpath:XPath Filter="subtract" xmlns:dsig-
                                 xpath="http://www.w3.org/2002/06/xmldsig-
                                 filter2">descendant::*[local-name()='ENSNBEHLayer'][last()-
                                 0]/*[local-name()='Signature']</dsig-xpath:XPath>

                        </ds:Transform>

                        <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"
                           xmlns:ds="http://www.w3.org/2000/09/xmldsig#"/>

                  </ds:Transforms>

                  <ds:DigestMethod …/>

                  <ds:DigestValue> …</ds:DigestValue>

            </ds:Reference>

      </ds:SignedInfo>

      <ds:SignatureValue …>…</ds:SignatureValue>
```

```
    <ds:KeyInfo …>
    ...
    </ds:KeyInfo>
</ds:Signature>
```

Bei dieser Variante wird mit dem ersten <dsig-xpath.XPath…>-Ausdruck das oberste Exemplar eines Layers bestimmten Namens (hier ENSNBEH-Layer) ausgewählt. Um das oberste Layer diesen Namens zu erhalten, muss die Anzahl der in diesem Layer insgesamt enthaltenen Layer gleichen Namens von last() abgezogen werden (hier last() – 0, das interessierende Layer ENSNBEHLayer enthält also keine weiteren Layer des Namens ENSNBEHLayer).

Mit dem zweiten XPath-Ausdruck werden die vorhandenen Signaturen im Layer aus diesem ausgeklammert. Im Resultat werden hier also zwei unabhängig nebeneinander stehende Signaturen angebracht. Die zweite Signatur signiert die erste nicht mit.

Die hier vorgestellten Varianten stellen nur zwei mögliche Wege dar. Weitere Varianten, welche ebenfalls dafür sorgen, dass zwei Signaturen parallel nebeneinander dasselbe Layer oder Dokument signieren, ohne sich gegenseitig zu zerstören, sind möglich und in gleicher Weise gültig, wie die beiden hier vorgestellten.

4.4.3 Qualifizierte vs. Fortgeschrittene elektronische Signatur

4.5 Verwaltungsinformationen

4.6 Layer vs. Sichten

4.7 Pflichtfelder

5 Spezieller Aufbau der Dokumenttypen

6 Referenz

7 Prüfziffern

8 Anhänge