Technische Dokumentation (TDoc)

Einheitliche Datenschnittstelle XML Authentifizierung

Bezeichnung	*Verfahren:* *ELSTER* *Produkt:* *Einheitliche Elster-Datenschnittstelle XML Authentifizierung* Auftragnehmer: BayLfSt - Dienststelle München
Verfahrensmanager	Roland Krebs
Produktmanager	Marcus Scherz
Dokumentverantwortlicher	Marcus Scherz
Version	2.1.0
Erstellt am	26.03.2010
Zuletzt geändert	17.03.2017 09:13
Bearbeitungszustand	☐ in Bearbeitung ☐ Vorgelegt ☒ fertig gestellt

Änderungsnachweis

Änderung		Geänderte Kapitel	Beschreibung der Änderung	Autor	Zustand
Datum	Version				
26.03.10	0.0.1	Alle	Ausgliederung aus dem Dokument „EBA-XML-Schnittstelle_3.6.4.doc“	Michael Stoll	Fertig gestellt
08.04.10	1.0		Review Andrea Maier	Michael Stoll	Fertig gestellt
04.06.12	1.1	2.1.3	Ripemd-160 entfernt	Michael Stoll	Fertig gestellt
05.12.12	1.1.1	2.1.3	Warnhinweis bzgl. Dokumenten, die nicht dieser Spezifikation entsprechen X509Data darf jetzt mehrere X509 Zertifikate enthalten	Michael Stoll	Fertig gestellt
12.06.13	1.1.2	2.1.2	Korrekturen/Ergänzungen bei Punkten a), c)	Matthias Wurm	Fertig gestellt
29.07.15	2.0.0	1, 2, 3	Aktualisierungen in Bezug auf Einführung von Signaturen mittels RSASSA-PSS	Alexander Maringer	Fertig gestellt
17.03.17	2.1.0	Alle	Aktualisierung des ELSTER-XML-Namespaces auf http://www.elster.de/elsterxml/schema/v11 ; Nutzdaten-Signaturen nicht mehr unterstützt; Anpassung der PKCS#1V1.5 –Frist gemäß BSI-Algorithmenkatalog 2017.	Matthias Wurm	Fertig gestellt

Inhaltsverzeichnis

Abbildungsverzeichnis

1 Einleitung

Das ElsterXML muss komprimiert, verschlüsselt und je nach Fachverfahren ggf. authentifiziert werden. Für die Authentifizierung wird die XML Signatur nach w3c verwendet.

In diesem Dokument wird die Signatur von Elster XML Dokumenten (entsprechend der einheitlichen XML-Datenschnittstelle für alle ELSTER-Verfahren) beschrieben.

Im Zuge der Umstellung von Signatur- und Verschlüsselungsalgorithmen auf PKCS#1v2.1 wird im Folgenden die Signaturerstellung mittels PKCS#1v2.1 (RSASSA-PSS) nach RFC 6931 ergänzt.

Hinweise sind in einem grünen Kasten hervorgehoben.

Wichtige Änderungen zum vorhergehenden Dokument sind gelb markiert.

Die einzelnen Felder der ElsterSignatur werden in diesem Dokument erläutert, sowie ggf. Formate festgelegt und mögliche Inhalte vorgegeben. Desweiteren ist die Struktur des ElsterXMLs dargestellt.

1.1 Übersicht Datenaufbau

Es wird ein zum Teil verschlüsselter XML-Datensatz an die Server der Steuerverwaltung übermittelt. Dieser besteht aus TransferHeader (THeader bzw. TH), NutzdatenHeader (NHeader bzw. NDH) sowie einer anwendungsspezifischen Datenstruktur (Nutzdaten, NDS) (Abb. 1).

Der TransferHeader ist der XML-Datenteil, der weitestgehend unverschlüsselt bleibt. Er wird dem verschlüsselten Teil des XML-Datensatzes vorangestellt. Der TransferHeader enthält wichtige Informationen für die Verarbeitung der Daten in der Clearingstelle und für die Verteilung in die Bundesländer, außerdem kann er Rückgabemeldungen und Fehlermeldungen aufnehmen.

Innerhalb des verschlüsselten Teils des XML-Datensatzes befinden sich sowohl der NutzdatenHeader, der alle Informationen für die Verarbeitung des Datensatzes, sowie Rückgabemeldungen und Fehlermeldungen aufnehmen kann, als auch der eigentliche datentypabhängige Datensatz (Nutzdatensatz).

Es können grundsätzlich ein oder - bei Sammellieferungen - mehrere Nutzdatenblöcke vorkommen.

Mit der Einführung der TransferHeader-Version 11 existiert nur noch eine Art der Authentifizierung, diese erfolgt über den verschlüsselten, komprimierten, base64-Kodierten DatenTeil und wird im TransferHeader abgelegt (Vorgang: „send-Auth“).

Die Signatur im NutzdatenHeader (Vorgang: „send-Sig“) wird nicht mehr unterstützt.

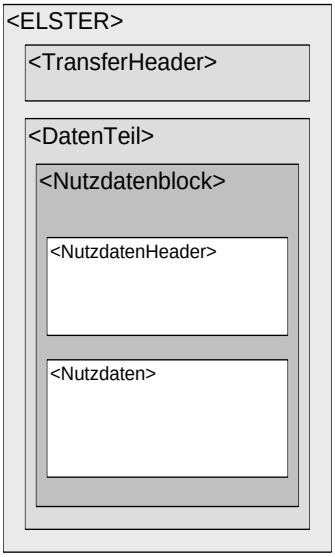

Abb. 1: ElsterXML-Daten

Beispiel:

- Eine Anfrage wird geschickt, das Ergebnis muss zu einem späteren Zeitpunkt abgeholt werden.
- Lohnsteuerbescheinigungen werden geliefert. Das Ergebnis, ob die Daten erfolgreich verarbeitet werden konnten, kann erst zu einem späteren Zeitpunkt abgefragt werden (LStB und Protokollanforderung->Protokoll).

2 Standards und Technik

2.1 TH-Authentifizierung (Signaturdaten werden im TransferHeader abgelegt)

Die TH-Signatur wird über das Element „DatenTeil“ und dessen verschlüsselten, komprimierten, Base64-kodierten Inhalt gebildet. Die Signatur und die zugehörigen Informationen werden im TransferHeader unter <SigUser> erwartet. Als Vorgang muss im TransferHeader angegeben werden: <Vorgang>send-Auth</Vorgang.

Kurzer Überblick zur Authentifizierung mit Vorgang „send-Auth“

a) Komprimierung (GZip), Verschlüsselung und Base64-Kodierung des Inhalts des Tags <DatenTeil>.

b) Signaturerstellung über das Tag <DatenTeil> und dessen Inhalt. Die Signatur wird im TransferHeader abgelegt unterhalb von <SigUser> im Tag <Sig>.

c) Der Inhalt des <SigUser>-Tags im TransferHeader muss ebenso wie der Inhalt des Tags <DatenTeil> komprimiert, verschlüsselt und Base64-kodiert werden, da es datenschutzrechtliche Inhalte enthält. Bei der Rückantwort vom Server wird das Tag <SigUser> nicht mehr übermittelt.

2.1.1 Profilierung der Signatur

Eine W3C-konforme Signatur kann in wesentlichen Bereichen, wie der Syntax, den verwendeten mathematischen Methoden und der Userinformation variiert werden, so dass die möglichen Formate gültiger W3C-Signaturen schier unendlich sind. Es ist im Rahmen der ElsterSignatur jedoch nicht notwendig, alle möglichen Kombinationen aus der Syntax und den verschiedenen Algorithmen zur Berechnung von Signatur und Digest zu zulassen. Darüber hinaus sind den Zertifikaten und deren Prüfung ebenfalls enge Grenzen gesetzt, so dass sich hierdurch eine weitere natürliche Eingrenzung ergibt.

Eine positive Behandlung von Dokumenten, die nicht dieser Spezifikation entsprechen, sondern nur w3c konform sind, kann durchaus möglich sein. Es entsteht jedoch kein Anspruch daraus, dass solche Dokumente auch in neueren Versionen positiv behandelt werden.

Es werden „detached Signatures“ verwendet, da die signierten Daten weder innerhalb der Signatur (signature is over content found within an Object element of the signature itself= enveloping) liegen noch die Daten die Signatur enthalten (signature is over the XML content that contains the signature as an element= enveloped), sondern vielmehr Daten referenzieren, die außerhalb der Signatur liegen und via URI oder XPath -Transform spezifiziert sind (is over content external to the Signature element, and can be identified via a URI or transform). Das stellt natürlich in keiner Weise eine Wertung dar, sondern dient lediglich der Vereinfachung und somit der Vermeidung von Fehlern.

Das Präfix „dsig“ soll zwingend verwendet werden.

Folgende Tabelle MUSS bei der Erstellung der Signatur Berücksichtigung finden.
Mit * gekennzeichnete Einträge sind Altlasten, diese werden aber aus Kompatibilitätsgründen noch unterstützt.

Element	**Erläuterung zu Attributen, Namespaces, Inhalt**
Signature	Id="Sign1" für die erste Signatur Id="Sign2" für die (zeitlich) zweite Signatur usw. Namespace Deklaration: xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"

CanonicalizationMethod	Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"
SignatureMethod	Algorithm="http://www.w3.org/2007/05/xmldsig-more#sha256-rsa-MGF1" oder Algorithm=http://www.w3.org/2007/05/xmldsig-more#rsa-sha256 *
Transform	Algorithm="http://www.w3.org/2002/06/xmldsig-filter2" Mit folgenden Präfixen (namespace prefix): xmlns:dsig-xpath="http://www.w3.org/2002/06/xmldsig-filter2" xmlns:elsterDefaultNs="http://www.elster.de/elsterxml/schema/v11" oder Algorithm="http://www.w3.org/TR/1999/REC-xpath-19991116" *
DigestMethod	Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"
KeyInfo	Enthält genau ein Element vom Typ X509Data Type="http://www.w3.org/2000/09/xmldsig#X509Data"
X509Data	Enthält - genau ein Element vom Typ „X509Certificate“ oder - eine Sequenz von Elementen vom Typ „X509Certificate“, dabei muss genau ein Zertifikat als Benutzerzertifikat identifizierbar sein (basicConstraints cA=false)
X509Certificate	Enthält das base64 kodierte X509v3 Benutzerzertifikat mit dem die Signatur erstellt wurde

Tabelle 1: Unterstützte Standards bei der XML-Signatur

2.1.1.1 Hash Algorithmen

Generell werden die SHA-2 Hashverfahren in folgenden Algorithmen

- Algorithm="http://www.w3.org/2007/05/xmldsig-more#sha256-rsa-MGF1"
- Algorithm="http://www.w3.org/2007/05/xmldsig-more#rsa-sha256Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"

für die Signatur von ElsterXML Dokumenten benutzt.

> Andere Hashverfahren als SHA-256 sind nicht zulässig.

2.1.1.2 Signatur Algorithmen

Generell werden die RSA PKCS#1 Signaturverfahren

- Algorithm="http://www.w3.org/2007/05/xmldsig-more#sha256-rsa-MGF1"
- Algorithm="http://www.w3.org/2007/05/xmldsig-more#rsa-sha256

für die Signatur von ElsterXML Dokumenten benutzt.

> RSA Signatur nach PKCS#1V1.5 ist noch bis Ende 2017 zulässig

2.1.1.3 Transform Algorithms

2.1.1.3.1 XPath Filter 2.0

Die „XPath Filter 2.0“ Transformation soll zur Bildung der Knotenmenge, über die signiert wird, verwendet werden. Das XPath Element in der Transformation hat den namespace „http://www.w3.org/2002/06/xmldsig-filter2", es wird empfohlen, das Präfix „dsig-xpath“ zu verwenden:

- xmlns:dsig-xpath="http://www.w3.org/2002/06/xmldsig-filter2"

Für die Signatur von ElsterXML MUSS ein namespace Präfix für den Elster default namespace definiert werden. Es wird empfohlen, das Präfix „elsterDefaultNs“ zu verwenden:

- xmlns:elsterDefaultNs="http://www.elster.de/elsterxml/schema/v11"

Für die Signatur über den Datenteil wird festgelegt:
Das Attribut „Filter“ hat den Wert „intersect“ und der XPath Ausdruck muss mit dem oben festgelegten namespace Präfix angegeben werden:

- //elsterDefaultNs:Datenteil

Folgende Transform Elemente sind zu diesen Festlegungen konform:

```
<dsig:Transform
   Algorithm="http://www.w3.org/2002/06/xmldsig-filter2">
   <dsig-xpath:XPath
      xmlns:dsig-xpath="http://www.w3.org/2002/06/xmldsig-filter2"
      xmlns:elsterDefaultNs="http://www.elster.de /elsterxml/schema/v11"
      Filter="intersect">
      //elsterDefaultNs:DatenTeil
   </dsig-xpath:XPath>
</dsig:Transform>
```

Einschränkungen:
Die maximale Größe des zu signierenden verschlüsselten Datenteils eines ElsterXML ist 5MByte.

Wird die Größenbeschränkung erreicht, müssen neue Varianten für die XML Signatur definiert werden.

2.1.1.3.2 XPath Filtering

Die Transformation mit „XPath Filtering“ ist die veraltetete Form für die ElsterSignatur. Sie wird aus Kompatibilitätsgründen noch unterstützt. Neue Implementierungen und Software Releases müssen die „XPath Filter 2.0“ Transformation verwenden.

Die Transformation mit „XPath Filtering“ wird bei der ElsterSignatur durch die „XPath Filter 2.0“ Transformation ersetzt.

Für die Signatur über den Datenteil wird festgelegt:
Der XPath Ausdruck wird ohne namespace Präfix angegeben:

- ancestor-or-self::Datenteil

Der Default-Namespace (xmlns="http://www.elster.de/elsterxml/schema/v11") muss bei der Kanonisierung vor der Signaturbildung herausgenommen werden, sonst kann der XPath Ausdruck nicht korrekt ausgewertet werden (leere Knotenmenge für die Hashbildung).

Folgende Transform Elemente sind zu diesen Festlegungen konform:

```
<dsig:Transform
  Algorithm="http://www.w3.org/TR/1999/REC-xpath-19991116">
  <dsig:XPath>
     ancestor-or-self::DatenTeil
  </dsig:XPath>
</dsig:Transform>
```

2.2 *Datenteil –Signatur der verschlüsselte Daten*

Beispiel einer Umsatzsteuervoranmeldung:

Verschlüsselt ist der Inhalt von „DatenTeil“, ohne das Tag „DatenTeil“ selbst.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Elster xmlns="http://www.elster.de /elsterxml/schema/v11">
   <TransferHeader version="11">
     <Verfahren>ElsterAnmeldung</Verfahren>
     <DatenArt>UStVA</DatenArt>
     <Vorgang>send-Auth</Vorgang>
     <Testmerker>700000004</Testmerker>
     <SigUser>
        <Sig>
           <dsig:Signature
              xmlns:dsig="http://www.w3.org/2000/09/xmldsig#" Id="Sign1">
              <dsig:SignedInfo>
                 <dsig:CanonicalizationMethod
                    Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315" />
                 <dsig:SignatureMethod
                    Algorithm="http://www.w3.org/2007/05/xmldsig-more#sha256-rsa-
MGF1" />
                 <dsig:Reference URI="">
                    <dsig:Transforms>
                       <dsig:Transform
                          Algorithm="http://www.w3.org/2002/06/xmldsig-filter2">
                          <XPath
                             xmlns="http://www.w3.org/2002/06/xmldsig-filter2"
                             xmlns:elsterDefaultNs="http://www.elster.de
/elsterxml/schema/v11"
                             Filter="intersect">
                             //elsterDefaultNs:DatenTeil
                          </XPath>
                       </dsig:Transform>
                       <dsig:Transform
                          Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-
20010315" />
                    </dsig:Transforms>
                    <dsig:DigestMethod
                       Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
```

```
                <dsig:DigestValue>
                  oNMeDzF8NSLaeyDRgD4oXPfwGMPKQS+tr17qEV6XTvI=
                </dsig:DigestValue>
              </dsig:Reference>
            </dsig:SignedInfo>
            <dsig:SignatureValue>
              QPtWJjlwZEkRPrjbIqZn5/HigYe/4XUTM/le59xNvpBvs8kY/oUIBrcjcAkLbkSv
              RnxHt3XAhvr5CSUi1bEYyQr2YgEFEwnyWSNINbli4lxmqIwJfs0qybzHX8QBUGPS
              xA0mTVDay90SUGy4nS6k7HcoV52EhLJOWfcuF//u3kXlEgPBcwNFLuoyEFZqZvku
              LkdHl6WQsOuG9hFsXcipUIKzD5BZAfPEvqS+hW0DWhw35iU4dVz0yxG6Y+Sp1r9t
              EDFAEFBKo7kyMP+UZJXmyfR/wz3hc7GLtbYq8xYPOt5K8eLxM+PQ6KzSn2THH22X
              KnGw13KWD+W+DOj7eWFAMA==
            </dsig:SignatureValue>
            <dsig:KeyInfo>
              <dsig:X509Data>
                <dsig:X509Certificate>
MIIEUjCCAzqgAwIBAgIEO7XQezANBgkqhkiG9w0BAQsFADBCMQswCQYDVQQGEwJE
RTEPMA0GA1UEChMGRWxzdGVyMQswCQYDVQQLEwJDQTEVMBMGA1UEAxMMRWxzdGVy
U29mdENBMB4XDTA5MDMxNzE2MzgxMVoXDTEyMDMxNzE2MzgxMVowKzEUMBIGA1UE
BRMLMTAwMTgxNzk4NEExEzARBgNVBAMTCjEwMDE4MTc5ODQwggEiMA0GCSqGSIb3
DQEBAQUAA4IBDwAwggEKAoIBAQCp8RL8aeHjFYCrXy4XJxY+J7l8lh/HKiKUQ8FV
6SzrALyE/+w2U+RYHBJyw8Zwom+xV5k7S2af/JdtUjmi2wBjIXufAuPHN/qTH8d+
boVzP6YQcNaf/wRCGr0bcYi/dPkyeaOsoK+ZYMTLwzSfCFNck2Xx7cwntQw7xiHh
MbCeGu8WlImwJwVqud5KKpu55vtMYfWf40pmdJiYt/3Y0vPMk93nZNmx+ldi0R4I
6sWdCb2zLGdX7AtjZ0SuI7zFxXQT/RgwPr5tPZ9Aw7ZTS3aWChmsUE8d77/mUYG6
hai9Aq729ShdYTVUeFXwVnaDAyeepIjES+92jM1jVb9ikgrLAgMBAAGjggFlMIIB
YTAOBgNVHQ8BAf8EBAMCB4AwDAYDVR0TAQH/BAIwADAdBgNVHQ4EFgQU22FEG8n7
XvbPU6k0V6TpuUJGZWswSwYDVR0fBEQwQjBAoD6gPIY6aHR0cDovL2dkbWlkLXRl
c3RjYS5nZG1pZC50ZXN0L0NlcnRFbnJvbGwvZ2RtaWQtdGVzdGNhLmNybDAfBgNV
HSUEGDAWBgorBgEEAYI3FAICBggrBgEFBQcDAjBxBgNVHSMEajBogBTybe+uKbT5
sCBu+WiZqQ/PetsF9qFKpEgwRjELMAkGA1UEBhMCREUxDzANBgNVBAoTBkVsc3Rl
cjEPMA0GA1UECxMGUm9vdENBMRUwEwYDVQQDEwxFbHN0ZXJSb290Q0GCBDuayhQw
QQYDVR0RBDowOIESdGVzdGVyMUBnZG1pZC50ZXN0oCIGCisGAQQBgjcUAgOgFAwS
dGVzdGVyMUBnZG1pZC50ZXN0MA0GCSqGSIb3DQEBCwUAA4IBAQBgg5U7K8W0yO0m
ACXE92nGfPnhhB1yqXGgr1m5xD0dc2MiFK1gj7IL+/uDtJ7lFBHqccvONKIrGP/8
IvuMf847lnWuRingkJMj7NTBZG6cShItie0kRDvPNa3XKAYaeAzetHk3Y2ZX40j5
vmmlHYL/xzMHh7fVMnXUtCyej+1JhYxQZE7tWf9b+y9T/1zas1BuiQ9OyC0CE1MO
mvjSMTmYelAmYAb0VDUCdo3iDvY+saaoa0yVIKkUg+sfMV77vJh4zVca+rBrjsid
3oPpTkzPTQAPWxIsFusY5HhCe2bNknhraheTf+cJu1dyvwKMWfep38/40D43UMpz
HpUhhYAu
                </dsig:X509Certificate>
              </dsig:X509Data>
            </dsig:KeyInfo>
          </dsig:Signature>
        </Sig>
    </SigUser>
    <HerstellerID>74931</HerstellerID>
    <ISO_8859-1_spezifisch>äöüÄÖÜß</ISO_8859-1_spezifisch>
    <ISO_8859-15_spezifisch>¤¦¨´¸½¼¾</ISO_8859-15_spezifisch>
    <DatenLieferant>Name</DatenLieferant>
    <Datei>
      <Verschluesselung>CMSEncryptedData</Verschluesselung>
      <Kompression>GZIP</Kompression>
      <DatenGroesse>7742</DatenGroesse>
      <TransportSchluessel />
    </Datei>
    <VersionClient>COALA Version 3.5.0 - 16.11.2006</VersionClient>
    <Zusatz>
      <Info>test</Info>
    </Zusatz>
```

```
   </TransferHeader>
   <DatenTeil> MIAGCSqGSIb3DQEHA…. verschlüsselter Inhalt ….eO1vAAAAAAAAAAAAAA==
   </DatenTeil>
</Elster>
```

3 Abkürzungsverzeichnis

Abkürzung	Erklärung
NDH	NutzdatenHeader (s. 1.1)
TH	TransferHeader (s. 1.1)