



Zentrum für sichere Informationstechnologie – Austria Secure Information Technology Center – Austria

A-1040 Wien, Weyringergasse 35
Tel.: (+43 1) 503 19 63-0
Fax: (+43 1) 503 19 63-66

A-8010 Graz, Inffeldgasse 16a
Tel.: (+43 316) 873-5514
Fax: (+43 316) 873-5520

http://www.a-sit.at
E-Mail: office@a-sit.at

PROJEKTBERICHT - OASIS-DSS

DIGITAL SIGNATURE SERVICE

VERSION 1.0, 07. DEZEMBER 2007

DI Bernd Zwattendorfer – bernd.zwattendorfer@iaik.tugraz.at

DI Thomas Zefferer – thomas.zefferer@iaik.tugraz.at

Zusammenfassung: Mit der ständig zunehmenden Bedeutung von elektronischen Kommunikationsmedien vor allem auch für die Wirtschaft steigt ebenfalls die Relevanz der elektronischen Signatur, wobei in zunehmenden Maße für deren Erstellung und Verifikation auch Server-seitige Lösungen gefragt sind. Mit OASIS-DSS steht seit Kurzem ein Standard zur Verfügung, der die Verwendung eines serverbasierten Signaturservices spezifiziert.

Im Zuge dieses Projekts wurde dieser Standard analysiert und eine diesem Standard entsprechende Referenzimplementierung erstellt. Diese wurde schließlich auf einem A-SIT Server über ein ebenfalls in diesem Projekt entwickeltes Web-Interface zugänglich gemacht, um so ein praktisches und benutzerfreundliches Kennenlernen dieses neuen Standards für interessierte AnwenderInnen aus der Wirtschaft zu ermöglichen.

Die in diesem Projekt gewonnenen Erkenntnisse, sowie ein Überblick über die durchgeführten Arbeiten und die erzielten Resultate sind in diesem Bericht zusammengefasst.

Inhaltsverzeichnis

Inhaltsverzeichnis	1
Abbildungsverzeichnis	2
Executive Summary	3
1 Einleitung	4
2 OASIS DSS	7
2.1 DSS Signing Protocol	7
2.2 DSS Verifying Protocol	9
2.3 DSS Core Elements	10
2.4 DSS Core Bindings	10
2.4.1 HTTP POST Transport Binding	10
2.4.2 SOAP 1.2 Transport Binding	10
2.4.3 TLS Security Bindings	10
3 Referenzimplementierung	12
3.1 Architektur	12
3.1.1 DSS Bindings	12
3.1.2 DSS Library	13
3.1.3 DSS Configuration	14
4 Demo-Installation	15
4.1 Signaturerstellung	15
4.2 Signaturverifikation	17
5 Zusammenfassung	19
Glossar	20
Referenzen	21
Historie	22

Abbildungsverzeichnis

Abbildung 1 – Funktionsweise der elektronischen Signatur	5
Abbildung 2 – Server-seitige Verarbeitung von elektronischen Signaturen	5
Abbildung 3 – Architektur des DSS-Servers	12
Abbildung 4 - Aufbau der Transportnachricht	13
Abbildung 5 - DSS-Library Architektur	14
Abbildung 6 - Schematischer Aufbau Demo-Installation.....	15
Abbildung 7 - Screenshot SignRequest	16
Abbildung 8 - Screenshot SignResponse	16
Abbildung 9 - Screenshot VerifyRequest	17
Abbildung 10 - Screenshot VerifyResponse	18

Executive Summary

Für die Durchführung dieses Projekts wurden die folgenden Projektziele, auf deren Erlangen in diesem Dokument noch ausführlich eingegangen wird, vereinbart.

- a) Analyse OASIS-DSS
- b) Umsetzung der Kernfunktionen
- c) Bereitstellung einer Referenzumsetzung am Demoserver
- d) Umsetzung wesentlicher DSS Profile
- e) Einbringen in die Standardisierung DSS-X über OASIS-Mitgliedschaft

Zu Beginn des Projekts wurde der Standard OASIS-DSS eingehend analysiert um ein tiefgehendes und für die folgende Umsetzung auch notwendiges Verständnis der Spezifikation zu erlangen. Ein Überblick über die durch den OASIS-DSS Standard festgelegte Funktionalität und die verwendeten Elemente und Protokolle ist in Abschnitt 2 gegeben.

Basierend auf den Resultaten der Analyse des Standards wurden dessen Kernfunktionen schließlich in Form einer Referenzimplementierung umgesetzt. Die Struktur sowie weitere Details zur implementierten Umsetzung der Spezifikation sind in Abschnitt 3 dieses Dokuments zusammengefasst.

Nach der Implementierung der Kernfunktionalität, wurde die Referenzumsetzung an einem Demoserver installiert. Um eine einfache Verwendung des Signatureservices zu gewährleisten, wurde zudem ein entsprechender DSS-Client entwickelt, der eine graphische Benutzerschnittstelle zur Verfügung stellt. Damit ist es sehr einfach, die auf dem Demoserver installierte Referenzimplementierung des OASIS-DSS Standards zu nutzen. Eine Beschreibung der erstellten Demo-Installation und des entwickelten Web-Interfaces ist in Abschnitt 4 gegeben.

Durch die Implementierung der entsprechenden optionalen Elemente, welche für die Erstellung eigener Profile verwendet werden können, ist es zudem sehr einfach, die entwickelte Implementierung auf anwendungsspezifische Anforderungen anzupassen.

Durch die in diesem Projekt durchgeführten Arbeiten, konnte eine Referenzimplementierung des OASIS-DSS Standards entwickelt und bereitgestellt werden. Auf die erstellte Umsetzung kann über das zum Kennenlernen des Services entwickelte Web-Interface unter folgender URL zugegriffen werden.

http://demo.a-sit.at/el_signatur/dss/index.html

Über dieses Web-Interface können zu signierende Daten an das Service übermittelt werden. Diese Daten werden signiert und das Ergebnis dieser Signaturerstellung schließlich zurück an die BenutzerIn gesendet. Andererseits kann das Service auch dazu verwendet werden, erhaltene Signaturen auf deren Gültigkeit zu überprüfen. Auch hier können die entsprechenden Daten sehr einfach über das zur Verfügung gestellte Web-Interface übermittelt werden.

Für interessierte AnwenderInnen aus der Wirtschaft ergibt sich so die Möglichkeit, benutzerfreundlich und einfach eine standardisierte Server-seitige Lösung zur Erstellung und Verifikation elektronischer Signaturen kennenzulernen.

Durch die Entwicklung der Referenzimplementierung konnten wichtige Erkenntnisse über den OASIS-DSS Standard gewonnen werden. Diese Erkenntnisse konnte man durch eine OASIS-Mitgliedschaft sowohl in die Wartung des Standards, als auch in dessen Weiterentwicklung (DSS-X) einfließen lassen.

1 Einleitung

Das Erbringen einer Unterschrift oder Signatur stellt in vielen Bereichen des täglichen Lebens – sowohl privat als auch beruflich – ein oft verwendetes Mittel dar, um den persönlichen Willen zu bekunden. Mit einer persönlichen Unterschrift können so zum Beispiel Verträge abgeschlossen, Identitätsnachweise erbracht, oder Beglaubigungen ausgestellt werden. Das Instrument der Signatur ist heutzutage aus der modernen Gesellschaft nicht mehr wegzudenken.

Mit der zunehmenden Verbreitung neuer Informations- und Kommunikationstechnologien auch in vielen Bereichen der Wirtschaft und einer damit verbundenen Erweiterung der Kommunikationsmöglichkeiten stieg auch das Bedürfnis, ein elektronisches Pendant zur herkömmlichen rechtsgültigen Unterschrift zur Verfügung zu haben. Mit dem Bundesgesetz über elektronische Signaturen, BGBl. I Nr. 190/1999 idF BGBl. I Nr. 152/2001 (Signaturgesetz – SigG), welches die Richtlinie des Europäischen Parlamentes und des Rates vom 13.12.1999 über gemeinschaftliche Rahmenbedingungen für elektronische Signaturen (Signaturrichtlinie) umsetzt, steht in Österreich eine rechtliche Grundlage zur Verfügung, mit der seit Inkrafttreten dieses Gesetzes elektronische Signaturen der herkömmlichen Unterschrift gleichgestellt sind.

Eine elektronische Signatur ist dabei wie folgt definiert:

„elektronische Signatur: elektronische Daten, die anderen elektronischen Daten beigefügt oder mit diesen logisch verknüpft werden und die der Authentifizierung, also der Feststellung der Identität des Signators, dienen;“ [Ref01]

Eine elektronische Signatur hat also den Zweck die Authentizität und Integrität eines signierten Dokuments sicherzustellen. Das bedeutet, dass einerseits die Identität der UnterzeichnerIn im Nachhinein prüfbar ist, sowie andererseits auch sichergestellt wird, dass das Dokument nach Aufbringen der Signatur nicht mehr geändert werden kann, ohne die Gültigkeit der elektronischen Signatur zu brechen.

Zur Gewährleistung dieser Anforderungen kommen bekannte Verfahren der Kryptographie, im Speziellen jene der asymmetrischen Kryptographie, zum Einsatz. Die Besonderheit des asymmetrischen kryptographischen Ansatzes stellt die Verwendung eines Schlüsselpaares dar. Ein von der UnterzeichnerIn geheim zu haltender privater Schlüssel wird dabei für die Signaturerstellung herangezogen. Der entsprechende für die Signaturverifikation nötige Schlüssel wird allen potentiellen verifizierenden Parteien über eine Public-Key-Infrastructure (PKI) zur Verfügung gestellt. Aufgrund der verwendeten zugrundeliegenden mathematischen Funktionen ist es für die verifizierende Partei zwar möglich, mit Hilfe des veröffentlichten Schlüssels die Korrektheit der elektronischen Signatur zu überprüfen, jedoch kann mit diesem Schlüssel alleine keine gültige Signatur der ursprünglichen UnterzeichnerIn produziert werden. Ebenso ist es unmöglich vom veröffentlichten Schlüssel zur Signaturverifikation auf den zur Signaturerstellung verwendeten geheimen Schlüssel der UnterzeichnerIn Rückschlüsse zu ziehen.

Abbildung 1 illustriert den Vorgang der Signaturerstellung bzw. deren Verifikation. In die Erstellung der Signatur bei der AbsenderIn der Nachricht geht neben deren privaten Schlüssel auch das Dokument selbst ein. Dadurch wird sichergestellt, dass das Dokument nachträglich nicht mehr geändert werden kann ohne letztendlich auch die Gültigkeit der Signatur zu zerstören. Wie bereits erwähnt, muss neben der Integrität des Dokuments auch die Authentizität der UnterzeichnerIn durch die elektronische Signatur gewährleistet werden. Daher muss der zur Signaturverifikation verwendete öffentliche Schlüssel eindeutig an die Identität der UnterzeichnerIn gebunden sein. Diese Zuordnung wird durch den Einsatz von Zertifikaten für die SignaturprüferIn verifizierbar sichergestellt. Prinzipiell gibt es verschiedene Arten von Zertifikaten, welche sich durch unterschiedliche rechtliche Anforderungen, die Garantie des Sicherheitsniveaus des Ausstellungsprozesses und der Vertrauenswürdigkeit der AusstellerIn unterscheiden. Gemein ist allen Arten von Zertifikaten, dass sie eine eindeutige Verbindung zwischen der Identität der ZertifikatinhaberIn und deren verwendeten öffentlichen Schlüssel festlegen.

Weitere Informationen über die Verwendung der elektronischen Signatur in Österreich können von [Ref01] bezogen werden.

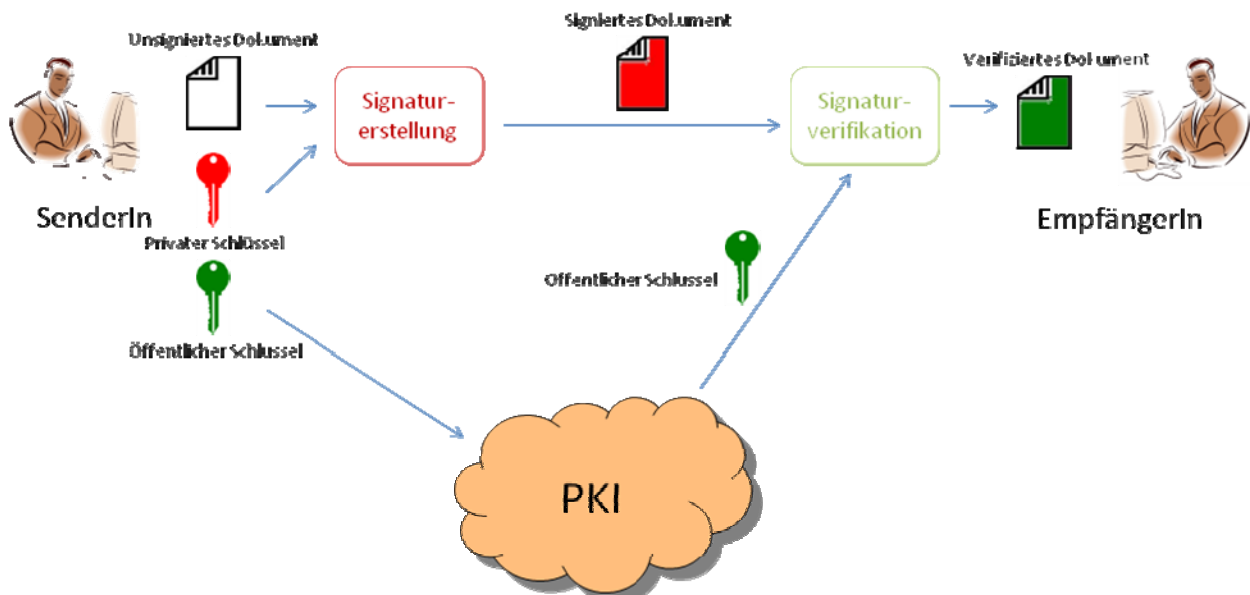


Abbildung 1 – Funktionsweise der elektronischen Signatur

Aufgrund der Vorteile, die ein Einsatz von elektronischen Signaturen mit sich bringt, erfreut sich diese auch in der Wirtschaft immer größerer Beliebtheit. Vor allem der Umgang mit Dokumenten, welche ausschließlich in digitaler Form vorliegen und von MitarbeiterInnen und PartnerInnen unabhängig von lokalen Distanzen einfach unterzeichnet werden können, machen den Einsatz elektronischer Signaturen auch für Unternehmen zunehmend interessant. Abhängig vom spezifischen Szenario, in dem die elektronische Signatur verwendet werden soll, können hier auch Server-seitige Lösungen zur Erstellung und Verifikation elektronischer Signaturen durchaus von Vorteil sein.

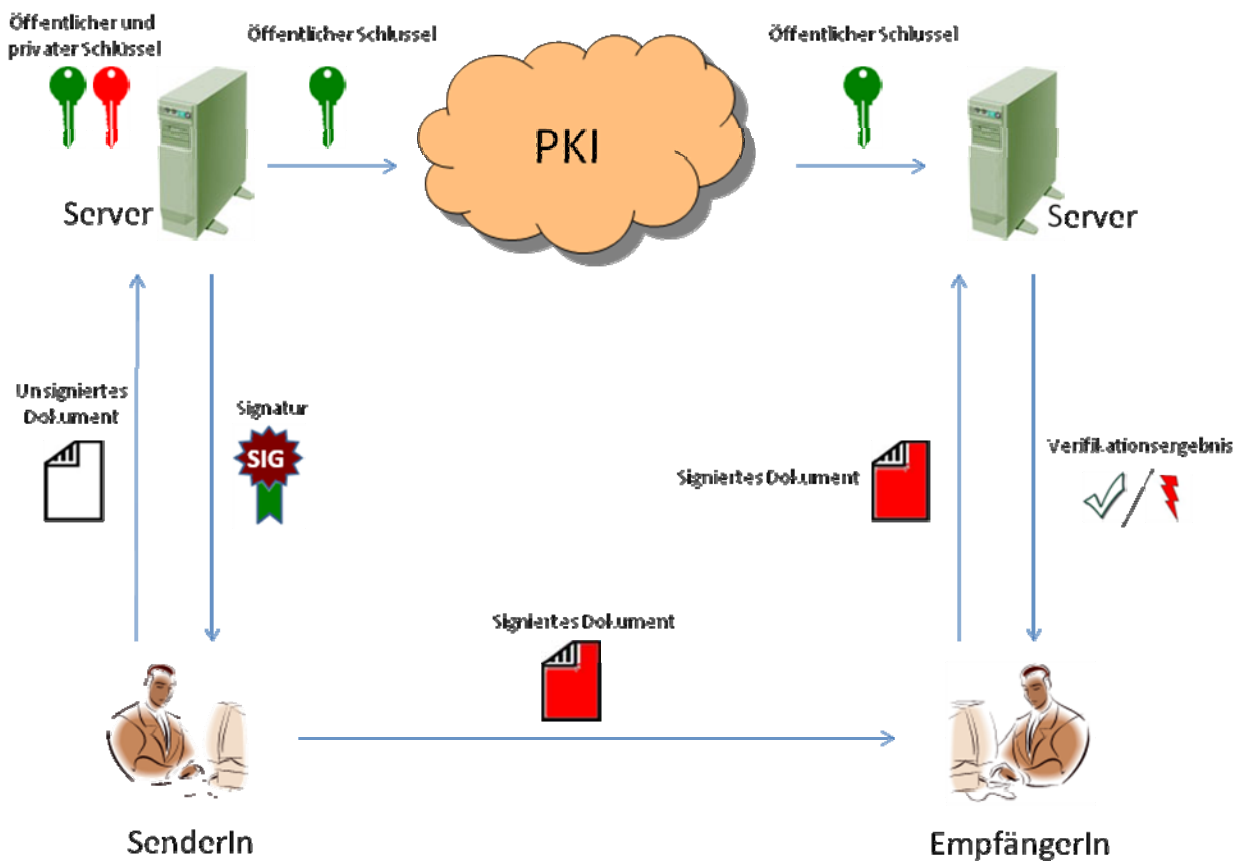


Abbildung 2 – Server-seitige Verarbeitung von elektronischen Signaturen

Abbildung 2 illustriert die prinzipielle Struktur einer Server-seitigen Infrastruktur zur Handhabung elektronischer Signaturen, sowie die Unterschiede zu herkömmlichen Verfahren.

Die AbsenderIn eines Dokuments, welches signiert werden soll, nimmt die Signaturerstellung nicht selbst vor, sondern übermittelt das zu unterzeichnende Dokument an einen Server, der die Signaturerstellung für die BenutzerIn durchführt. Dazu muss der entsprechende Server sowohl über den privaten Schlüssel der AbsenderIn, als auch über den korrespondierenden Schlüssel zur Signaturprüfung, der über eine PKI veröffentlicht wird, verfügen. Der berechnete Signaturwert des zu signierenden Dokuments wird daraufhin an die BenutzerIn zurückgeschickt.

Die EmpfängerIn führt nach Erhalt des signierten Dokuments dessen Verifikation nicht selbst durch, sondern übermittelt es an einen entsprechenden Server, welcher die Prüfung der aufgetragenen Signatur vornimmt. Das Ergebnis der Signaturverifikation wird der EmpfängerIn schließlich mitgeteilt, woraufhin diese entscheiden kann, wie sie mit dem nun geprüften Dokument weiter verfährt.

Das in Abbildung 2 skizzierte Szenario geht davon aus, dass sowohl UnterzeichnerIn als auch PrüferIn eine Server-seitige Lösung zur Handhabung von elektronischen Signaturen verwenden. Da die Signaturerstellung bzw. die Signaturprüfung prinzipiell zwei voneinander getrennte Verfahren sind, ist es ebenso gut möglich, dass nur eine der beiden Parteien auf eine Server-seitige Infrastruktur zur Verarbeitung von elektronischen Signaturen zurückgreift, während die andere Partei die Erstellung bzw. die Verifikation der Signatur ohne den Einsatz eines entsprechenden Servers vornimmt (z.B. lokale Erstellung einer qualifizierten Signatur mit Chipkarte und Prüfung über ein Server-seitiges DSS Modul).

Durch den Einsatz des in Abbildung 2 beschriebene Server-seitigen Ansatzes ergeben sich vor allem für Unternehmen mit einer wachsenden Anzahl von MitarbeiterInnen Vorteile gegenüber einer Client-seitigen Verarbeitung von elektronischen Signaturen. Wie in Abbildung 2 ersichtlich, wird das gesamte Schlüsselmanagement durch den Einsatz von Signaturservern von der BenutzerIn in die Server-Infrastruktur verlagert, wodurch sämtliche Schlüssel zentral verwaltet werden können. In Anbetracht der Tatsache, dass jede MitarbeiterIn, die elektronische Signaturen erstellen können soll, über ein entsprechendes individuelles Schlüsselpaar verfügen muss, wird klar, dass sich dadurch eine enorme verwaltungstechnische Effizienzsteigerung erzielen lässt. Wie bereits erwähnt, hängt die Sicherheit und Zuverlässigkeit der elektronischen Signatur von der Geheimhaltung des verwendeten privaten Schlüssels ab. Durch den Einsatz von Signaturservern und der daraus folgenden zentralen Schlüsselverwaltung ist es nicht mehr nötig, auf jedem Client einen privaten Schlüssel zu speichern. Die Sicherheit der verwendeten privaten Schlüssel hängt somit nur mehr von der Sicherheit eines einzelnen Servers und nicht mehr von jener aller verwendeten Clients ab. Da man im Allgemeinen davon ausgehen kann, dass ein Server professioneller gewartet wird als ein, von einer technisch vielleicht nicht versierten BenutzerIn verwendeter Client, trägt die Verwendung eines zentralen Signaturservers nicht nur zu Steigerung der Effizienz, sondern darüber hinaus auch zu einer Erhöhung der Sicherheit bei.

Mit OASIS' Digital Signature Service Core Protocols, Elements, and Bindings Version 1.0 (OASIS-DSS) [Ref02] steht seit kurzem ein Standard zur Verfügung, der die Verwendung eines Services zur Server-seitigen Signaturerstellung und Verifikation definiert.

Im Zuge dieses Projekts wurde dieser Standard analysiert, eine Referenzimplementierung erstellt, und diese als Web-Service installiert. Dieses Dokument gibt einen Überblick über die in diesem Projekt durchgeführten Arbeiten und fasst die erzielten Resultate zusammen.

2 OASIS DSS

Der OASIS-DSS Standard [Ref02] spezifiziert hauptsächlich zwei Protokolle zur Erstellung bzw. Verifikation von elektronischen Signaturen. Diese Protokolle basieren auf XML und erlauben es Clients, Anfragen zur Signaturerstellung bzw. Signaturprüfung an einen entsprechenden Server zu stellen.

Um die für unterschiedliche Anwendungsszenarien nötige Flexibilität aufweisen zu können, unterstützt der Standard die Verwendung von Profilen, welche auf die einzelnen Anwendungsfälle abgestimmt werden können. Hierfür definiert OASIS-DSS eine Vielzahl von sogenannten optionalen Inputs und Outputs, die zur Erstellung dieser Profile verwendet werden können. Optionale Elemente können beispielsweise die Art der zu erstellenden Signatur, der Zeitpunkt für den eine Verifikation der Signatur vorgenommen werden soll, oder zusätzliche für die Verifikation nötige Daten wie zum Beispiel Zertifikate sein.

Zur Kommunikation zwischen Client und Server müssen gängige Protokolle verwendet werden. Der Standard legt daher auch fest, wie OASIS-DSS in Verbindung mit einigen dieser Protokolle verwendet werden kann. Zusätzlich definiert der Standard einige zusätzliche Elemente, mit denen Funktionalität, die über die reine Verarbeitung von elektronischen Signaturen hinausgeht, implementiert und abgerufen werden kann.

2.1 DSS Signing Protocol

Das Protokoll, mit dem ein Client die Erstellung einer Signatur vom Server anfordern kann, spezifiziert zwei grundlegende Elemente, welche zum Austausch der nötigen Daten herangezogen werden.

Einerseits ist das Element `<SignRequest>`, das dazu verwendet wird, um die zu signierenden Daten und die für eine Signaturerstellung benötigten Informationen zum Server zu übertragen, folgendermaßen spezifiziert.

```
<xs:element name="SignRequest">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="dss:RequestBaseType"/>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
```

Der verwendete Typ `<RequestBaseType>` beschreibt die für eine Signaturerstellung relevanten Informationen und hat selbst die folgende Struktur.

```
<xs:complexType name="RequestBaseType">
  <xs:sequence>
    <xs:element ref="dss:OptionalInputs" minOccurs="0"/>
    <xs:element ref="dss:InputDocuments" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="RequestID" type="xs:string" use="optional"/>
  <xs:attribute name="Profile" type="xs:anyURI" use="optional"/>
</xs:complexType>
```

Das Element `<OptionalInputs>` enthält die durch das jeweilige Profil festgelegten zusätzlichen Informationen, die der Client an den Server übermittelt, während das Element `<InputDocuments>` die eigentlichen zu signierenden Daten enthält. Eine Definition dieser Elemente kann dem OASIS-DSS Standard [Ref02] entnommen werden.

Auf der anderen Seite enthält das Element `<SignResponse>` die Daten, welche nach der Verarbeitung einer entsprechenden Anfrage vom Server an den Client übermittelt werden und hat folgende Struktur.

```

<xs:element name="SignResponse">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="dss:ResponseBaseType">
        <xs:sequence>
          <xs:element ref="dss:SignatureObject" minOccurs="0"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>

```

Das verwendete Element `<SignatureObject>` enthält die berechnete Signatur des zu unterzeichnenden Dokuments. Die genaue Definition und Struktur dieses Elements kann ebenfalls dem OASIS-DSS Standard [Ref02] entnommen werden.

Das Element `<SignResponse>` besteht aus dem Typ `<ResponseBaseType>`, welcher wie folgt aufgebaut ist.

```

<xs:complexType name="ResponseBaseType">
  <xs:sequence>
    <xs:element ref="dss:Result"/>
    <xs:element ref="dss:OptionalOutputs" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="RequestID" type="xs:string" use="optional"/>
  <xs:attribute name="Profile" type="xs:anyURI" use="required"/>
</xs:complexType>

```

Eine genaue Definition sowie ausführliche Erläuterungen der in diesem Element verwendeten Elemente kann wiederum dem OASIS-DSS Standard [Ref02] entnommen werden.

Die exakte Vorgehensweise, nach welcher der Signaturserver nach Erhalt eines `<SignRequest>` Elements eine Signatur zu erstellen hat, ist ebenfalls im Standard definiert und gliedert sich im Falle einer zu erstellenden XML Signatur in die folgenden Schritte.

- 1) Für jedes Element `<Document>`, das im Element `<InputDocuments>` des `<SignRequest>` Elements enthalten ist, werden folgende Schritte ausgeführt.
 - a.) Die enthaltenen Daten werden ihrem Format entsprechend aus dem `<Document>` Element extrahiert.
 - b.) Die erhaltenen Daten werden verarbeitet und bestimmten Transformationen unterworfen, um eine kanonisierte Zeichenkette bestehend aus Achtbitzeichen zu erhalten.
 - c.) Über diese kanonisierten Daten wird ein Hash-Wert berechnet.
 - d.) Der Server erstellt ein `<ds:Reference>` Element, welches selbst wiederum eine Menge von Attributen und Elementen enthält, darunter auch den berechneten Hash-Wert.
- 2) Resultate, welche der Verarbeitung von optionalen Inputs entstammen, werden in die weitere Vorgehensweise ebenfalls eingebunden.
- 3) Der Server erstellt eine XML Signatur unter Verwendung der erhaltenen `<ds:Reference>` Elemente.

Abhängig von der Art des gesendeten `<SignRequest>` Elements unterscheiden sich die vom Server durchgeführten Aufgaben geringfügig. Beispielsweise entfallen die Punkte 1.a, 1.b und 1.c wenn anstatt zu signierender Daten vom Client bereits ein Hash-Wert über diese Daten übermittelt wird. Ebenso unterscheidet sich die Erstellung einer CMS Signatur in bestimmten Punkten von der Erstellung einer XML Signatur.

Der OASIS-DSS Standard spezifiziert für das Protokoll zur Server-seitigen Generierung von Signaturen eine Reihe optionaler Inputs, auf die bei der Erstellung persönlicher Profile zurückgegriffen werden kann. Eine Auflistung und Beschreibung dieser optionalen Inputs kann der Spezifikation [Ref02] entnommen werden.

2.2 DSS Verifying Protocol

So wie das Protokoll, über welches die Erstellung einer elektronischen Signatur angefordert werden kann, spezifiziert auch das entsprechende Protokoll zur Anforderung einer Signaturverifikation zwei grundlegende XML Elemente, über welche die benötigten Daten kommuniziert werden.

Um eine Signaturverifikation vom Server anzufordern, verwendet der Client das Element `<VerifyRequest>`, dessen Struktur im Folgenden gegeben ist.

```
<xs:element name="VerifyRequest">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="dss:RequestBaseType">
        <xs:sequence>
          <xs:element ref="dss:SignatureObject" minOccurs="0"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
```

Dieses Element erbt dabei vom Typ `<RequestBaseType>`, dessen Struktur bereits in Abschnitt 2.1 vorgestellt wurde. Zusätzlich enthält `<VerifyRequest>` ein Element `<SignatureObject>`, welches seinerseits die zu verifizierende Signatur bzw. eine Referenz auf dieselbige enthält.

Das zweite grundlegende Element, das im Verifikationsprozess zur Anwendung kommt, ist das `<VerifyResponse>` Element, welches vom Typ `<ResponseBaseType>`, der ebenfalls bereits in Abschnitt 2.1 erläutert wurde, erbt und zusätzlich keine weiteren Attribute oder Elemente definiert.

```
<xs:element name="VerifyResponse" type="dss:ResponseBaseType"/>
```

Erhält ein Server von einem Client ein entsprechendes `<VerifyRequest>` Element, vollzieht er laut Standard folgende Schritte um die übermittelte Signatur zu prüfen.

- 1) Der Server extrahiert alle erhaltenen `<ds:Signature>` Elemente. Abhängig vom Format der übermittelten Daten werden dafür unterschiedliche Strategien verfolgt.
- 2) Für jedes `<ds:Reference>` Element eines `<ds:Signature>` Elements wird das entsprechende Input Dokument determiniert. Auch hier werden abhängig von der Form, in der dieses Dokument bereitgestellt wird, spezifische Strategien zur Erfüllung dieser Anforderung angewendet.
- 3) Abhängig von einem eventuell vorhandenen optionalen Input `<UseVerificationTime>` wird die Gültigkeit der Signatur zu einem bestimmten Zeitpunkt evaluiert.

- 4) Abhängig vom Ergebnis der Signaturverifikation liefert der Server neben einem `<ResultMajor>` Code auch einen entsprechenden `<ResultMinor>` Code zurück.

Der OASIS-DSS Standard definiert auch das Verhalten des Servers, wenn das übermittelte Input-Dokument mehrere Signaturen enthält, oder wenn ein Timestamp-Token verifiziert werden soll. Nähere Informationen dazu sind in der Spezifikation des Standards unter [Ref02] zu finden.

Des Weiteren ist dort festgelegt, wie eine Verifikation von CMS Signaturen, welche sich aufgrund der Besonderheiten dieser leicht von der Verifikation von XML Signaturen unterscheidet, durchzuführen ist.

Vergleichbar mit dem Protokoll zur Erstellung von Signaturen spezifiziert der Standard auch für deren Verifikation eine Reihe optionaler Inputs und Outputs, mit denen abhängig von der entsprechenden Anwendung verschiedene Funktionalität in den Verifikationsprozess eingebunden werden kann. So kann über diese optionalen Elemente beispielsweise der Server veranlasst werden, die vorliegende Signatur für einen bestimmten Zeitpunkt zu prüfen. Eine andere Möglichkeit besteht darin, dem Server über diese optionalen Elemente zusätzliche für die Signaturverifikation relevante Daten zu übermitteln.

2.3 DSS Core Elements

Neben den in den Abschnitten 2.1 und 2.2 beschriebenen Protokollen, über welche der Client eine Server-seitige Erstellung bzw. Verifikation von elektronischen Signaturen anfordern, bzw. die entsprechende Antwort des Servers entgegennehmen kann, spezifiziert der OASIS-DSS Standard auch noch eine Reihe von zusätzlichen Elementen, welche in Verbindung mit den beiden Hauptprotokollen verwendet werden können. Beispielsweise wird ein `<Timestamp>` Element definiert, welches von spezifizierten Profilen verwendet werden kann, um das Service um eine entsprechende Zeitstempelfunktionalität zu erweitern.

2.4 DSS Core Bindings

Unter dem Namen „DSS Bindings“ definiert der OASIS-DSS Standard Möglichkeiten, wie die Kommunikation zwischen Client und Server in gebräuchliche Kommunikationsprotokolle eingebunden werden kann. Darüber hinaus definieren sogenannte Security Bindings, wie Anforderungen betreffend Datenintegrität, Authentifizierung und Datendiskretion für die Client-Server Kommunikation unter Verwendung gängiger Protokolle erreicht werden können.

2.4.1 HTTP POST Transport Binding

Dieses Binding definiert die Client-Server Kommunikation über HTTP [Ref03]. Dazu wird vom Standard festgelegt, wie eine entsprechende Anfrage des Clients an den Server auszusehen hat. Im Speziellen werden unter anderem die Werte, welche die Kopfzeilen der HTTP Anfrage annehmen müssen, spezifiziert.

Dasselbe gilt für die HTTP Antwort des Servers. Auch hier ist festgelegt, wie diese auszusehen hat, um dem Standard zu genügen.

2.4.2 SOAP 1.2 Transport Binding

Neben HTTP ist im OASIS-DSS Standard auch die Verwendung von SOAP 1.2 [Ref04] zum Austausch von Nachrichten zwischen Server und Client spezifiziert. Wiederum wird eine Reihe von Regeln festgelegt, welche eingehalten werden müssen um eine korrekte Verwendung von DSS in Verbindung mit diesem Nachrichtenprotokoll zu gewährleisten. Die Verwendung des SOAP Attachment Features [Ref05] wird dabei ebenfalls unterstützt und ist im OASIS-DSS Standard spezifiziert.

2.4.3 TLS Security Bindings

TLS [Ref07] ist ein Protokoll, das Lösungen für die kryptographischen Anforderungen Datenintegrität, Authentifizierung und Datendiskretion anbietet. OASIS-DSS spezifiziert die Verwendung dieses Protokolls in Verbindung mit den in Abschnitt 2.4.1 und 2.4.2 beschriebenen Bindings, wodurch diese um die erwähnte kryptographische Funktionalität ergänzt werden können.

Im Speziellen werden diverse Security Bindings definiert, die diverse von TLS unterstützte Methoden der Authentifizierung spezifizieren. Details zu diesen Security Bindings können wiederum dem OASIS-DSS Standard [Ref02] entnommen werden.

3 Referenzimplementierung

Neben der Analyse des DSS-Standards stellte die Umsetzung dessen wichtigster Kernfunktionalitäten in Form einer Referenzimplementierung ein weiteres Ziel dieses Projekts dar. Diese Referenzimplementierung wird als Service am Demoserver von A-SIT [Ref07] zur Verfügung gestellt. Interessierten AnwenderInnen aus der Wirtschaft wird es dadurch ermöglicht, diesen neuen Standard für Server-seitige Signaturen einfach und praktikabel zu testen.

Dieser Abschnitt beschreibt den Aufbau und die Struktur sowie die verwendeten Komponenten der Referenzimplementierung. Da der DSS-Standard in seiner Spezifikation nicht sehr eng gestaltet ist und daher sehr viele Freiheiten für eine Implementierung zulässt, wurde der Schwerpunkt der Entwicklung auf die Basisfunktionalitäten gelegt. Erweiterungen im Rahmen der optionalen Inputs bzw. optionalen Outputs (siehe Abschnitt 2) wurden mittels entsprechender Schnittstellen vorgesehen, wodurch eine einfache Zugabe von weiteren Modulen (z.B. Profilmodule) ermöglicht wird.

3.1 Architektur

Abbildung 3 zeigt einen Überblick über die modulare Architektur des zur Verfügung gestellten DSS-Servers. Der DSS-Server besteht aus einer Library, welche die Kernfunktionalität implementiert (DSS-Library), sowie aus einem im DSS-Standard spezifizierten Binding (siehe Abschnitt 2.4), welches das Mapping zwischen DSS-Nachrichten und standardisierten Kommunikations- und Transportprotokollen übernimmt (DSS-Binding). Über eine Konfiguration können die Funktionen der DSS-Library einfach eingestellt und angepasst werden (DSS-Configuration).

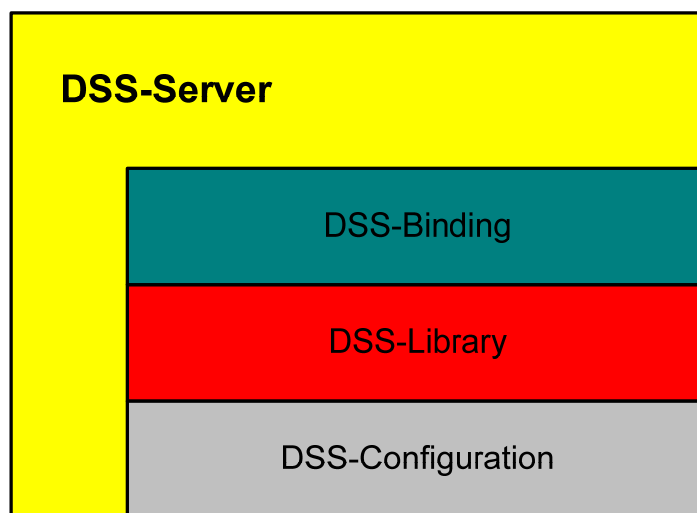


Abbildung 3 – Architektur des DSS-Servers

Sendet ein Client eine Anfrage an den DSS-Server, so wird diese vom DSS-Binding empfangen. Von dort wird die Anfrage an die DSS-Library weitergereicht, welche diese entsprechend den inkludierten Anweisungen abarbeitet. War die Abarbeitung erfolgreich, so wird eine entsprechende DSS-Response an das DSS-Binding übermittelt und an den Client zurückgesendet. Im Falle eines Misserfolgs bei der Abarbeitung wird eine aussagekräftige Fehlermeldung an den Client retourniert.

3.1.1 DSS Bindings

Unter dem Begriff Binding versteht man das Einbinden von OASIS-DSS Nachrichten in standardisierte Kommunikations- und Transportprotokolle. OASIS-DSS unterscheidet in seiner Spezifikation die Verwendung von Transportprotokollen (Transport Bindings) und Sicherheitsprotokollen (Security Bindings) [Ref02].

Im Rahmen der Referenzimplementierung wird das sogenannte SOAP 1.2 Transport Binding (siehe Abschnitt 2.4.2) verwendet. Das heißt, DSS-Nachrichten wie beispielsweise ein einzelner `<SignRequest>`, oder `<VerifyRequest>` bzw. die entsprechenden resultierenden Antworten werden in einer SOAP-Nachricht [Ref04] verpackt. Diese SOAP-Nachrichten werden mittels HTTP vom Client zum Server bzw. vom Server zum Client übertragen.

Die DSS-Referenzimplementierung steht am Demoserver von A-SIT als entsprechendes Web-Service unter folgenden URLs zur Verfügung.

- Signaturservice: <http://demo.a-sit.at/DSS/DSSSignService>
- Verifikationsservice: <http://demo.a-sit.at/DSS/DSSVerifyService>

Abbildung 4 zeigt den schematischen Aufbau von Transportnachrichten zwischen Client und Server. Eine DSS-Nachricht wird in den Body einer SOAP-Nachricht verpackt, welche anschließend mittels HTTP zwischen Client und Server übertragen wird.

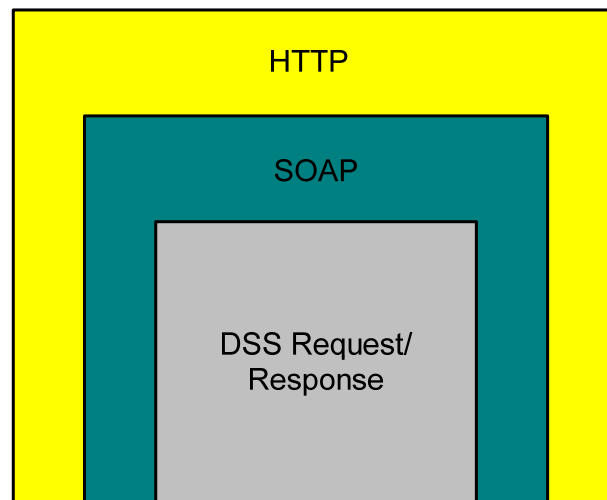


Abbildung 4 - Aufbau der Transportnachricht

Um eine sichere Übertragung dieser Nachrichten zu gewährleisten, kann dem Transport Binding noch ein Security Binding hinzugefügt werden. Im Falle der Referenzimplementierung kann dies einfach dadurch erreicht werden, indem statt des ungeschützten Transportprotokolls HTTP das sichere Übertragungsprotokoll HTTPS [Ref07] verwendet wird.

3.1.2 DSS Library

Die DSS-Library beinhaltet die Implementierung der Kernfunktionen der DSS-Spezifikation. Bei der Implementierung der Library wurde großer Wert auf Erweiterbarkeit gelegt, sodass Profile oder zusätzliche Funktionalität leicht und einfach hinzugefügt werden können, ohne die Implementierung wesentlich verändern zu müssen. Es wurden daher größtenteils Schnittstellen spezifiziert, die ein einfaches Hinzufügen von Modulen ermöglichen.

Die DSS-Library bildet also den Kern der Referenzimplementierung und ist für das Bearbeiten der DSS-Anfrage zuständig. Nach dem Empfang einer DSS-Nachricht über das DSS-Binding des DSS-Servers wird diese Nachricht an die DSS-Library weitergereicht. In der DSS-Library wird die Nachricht entsprechend der empfangenen Anfrage (inklusive optionaler Inputs) abgearbeitet.

Abbildung 5 zeigt den modularen Aufbau der prozessorientierten Architektur der DSS-Library. Der Core Processor bildet das Herzstück der Implementierung und verwaltet die einzelnen Unter-Prozessoren. Wird nun eine Anfrage empfangen, so teilt der Core Processor diese auf und delegiert die Daten oder Instruktionen an den entsprechenden Unter-Prozessor. Enthält nun beispielsweise eine DSS-Anfrage ein `<OptionalInput>` Element, so wird der Inhalt dieses Elements an den OptionalInput Processor weitergereicht. In diesem Unterprozessor werden die Daten bzw. Instruktionen dem Input entsprechend verarbeitet und je nach Inhalt die Zustände des Core Processors so gesetzt, dass die DSS-Anfrage wie gewünscht abgearbeitet werden kann. Im Prinzip versorgen die einzelnen Unterprozessoren den Core Processor mit Daten und

Instruktionen, damit dieser die gewünschte DSS-Response für die erhaltene DSS-Anfrage zusammenstellen kann.

Auch im Fall des Zusammenstellens einer DSS-Response Nachricht delegiert der Core Processor einzelne Aufgaben an Unter-Prozessoren weiter. Nachdem diese die ihnen übertragenen Aufgaben vollständig erledigt haben, stellt der Core Processor die DSS-Response der DSS-Anfrage entsprechend zusammen und reicht diese dem DSS-Binding zur Übermittlung an den Client weiter.

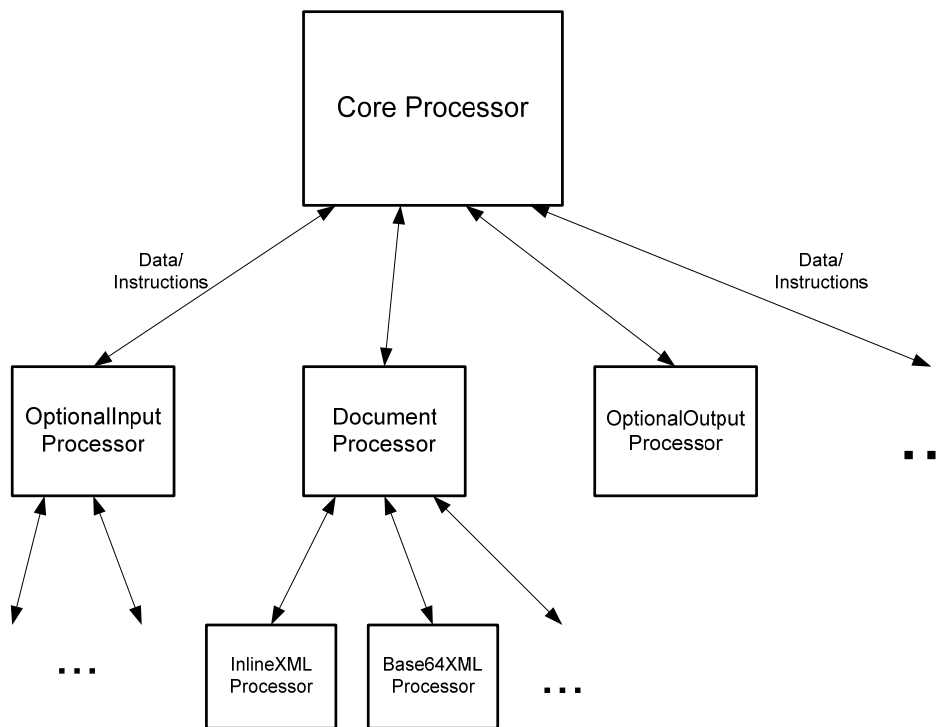


Abbildung 5 - DSS-Library Architektur

3.1.3 DSS Configuration

Die DSS-Configuration besteht aus einer XML-Datei, mit deren Hilfe die DSS-Library entsprechend konfiguriert werden kann. Die Konfigurations-Datei beinhaltet neben Positionen zur Vornahme von Einstellungen auch Positionen zur einfachen Erweiterung des DSS-Library. Soll beispielsweise ein Profil hinzugefügt werden, so kann hier die implementierende Klasse spezifiziert werden.

Wesentliche Konfigurationsmöglichkeiten sind unter anderem:

- **Keystore-Konfiguration:** Hier wird angegeben, welchen Keystore mit privaten Schlüsseln die DSS-Library zum Signieren und Verifizieren verwenden soll.
- **Methode zur Signaturerstellung:** Spezifizierung des Algorithmus zur Signaturerstellung, z.B. „dsa-sha1“
- **Methode zur Kanonisierung:** Bevor das Dokument signiert wird, wird es in eine einheitliche Form übergeführt (Kanonisierung). Hier kann der entsprechende Algorithmus zur Kanonisierung angegeben werden, z.B. „REC-xml-c14n-20010315“
- **Modul-Erweiterungen:** Angabe von Modul-Packages zur Erweiterung der DSS-Library

4 Demo-Installation

Wie in Abschnitt 3 beschrieben, wird die DSS-Referenzimplementierung als Web-Service zur Verfügung gestellt. An dieses Service können nun über SOAP der OASIS-DSS Spezifikation entsprechende Anfragen zur Signaturerstellung bzw. Signaturverifikation gestellt werden. Um auch technisch nicht so versierten BenutzerInnen die Möglichkeit zu bieten, das Service in Anspruch zu nehmen, wird ein einfaches Web-Interface zum Testen des Services zur Verfügung gestellt. Eine BenutzerIn kann also einfach mittels Web-Browser die Möglichkeiten Server-seitiger Signaturen testen. Damit soll eine einfache und benutzerfreundliche Verwendung der Referenzimplementierung sichergestellt werden.

Abbildung 6 zeigt den schematischen Aufbau der Demo-Installation am A-SIT Demoserver.

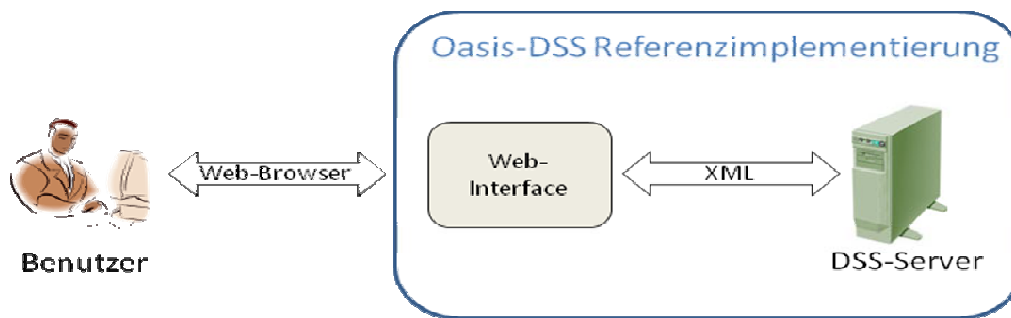


Abbildung 6 - Schematischer Aufbau Demo-Installation

Mit Hilfe des Web-Interfaces können XML-Dokumente Server-seitig signiert und entsprechend auch wieder Server-seitig verifiziert werden. Es bedarf also keines spezifischen Wissens über Web-Services, da die entsprechende DSS-Anfrage automatisch aus den eingegebenen Daten zusammengestellt wird. Die Erstellung und den Versand der Anfrage übernimmt ein DSS-Client, der vom Web-Interface aus angestoßen werden kann.

4.1 Signaturerstellung

Das Web-Interface und der darunterliegende DSS-Client ermöglichen die Erstellung von unterschiedlichen XML-Signaturen. Folgende Signatur-Typen werden dabei unterstützt:

- Detached Signature: Das signierte Dokument und die Signatur sind getrennt, die Signatur enthält nur eine Referenz auf die signierten Daten.
- Enveloped Signature: Hier wird die Signatur in das signierte Dokument eingebettet.
- Enveloping Signature: In diesem Fall umhüllt die Signatur die signierten Daten.

Das zu signierende Dokument kann entweder in reinem XML oder in Base64-codiertem XML angegeben werden. Zur Signaturerstellung wird jener private Schlüssel verwendet, der in der DSS-Library standardmäßig eingestellt ist.

Abbildung 7 zeigt einen Screenshot des zur Signaturerstellung verwendeten Web-Interfaces. Durch das Drücken des Knopfes „Signieren“ wird ein DSS-SignRequest zusammengestellt und an den DSS-Server gesandt. Der DSS-Server verarbeitet die Abfrage und schickt die entsprechende DSS-Response Nachricht an den Client zurück. Der Client filtert die Response und stellt das entsprechende Resultat (siehe Abschnitt 2), sowie die erstellte Signatur im Browser des Benutzers dar. Im Fehlerfall wird nur die Fehlermeldung des DSS-Servers angezeigt. Die vom Server gesendete DSS-Response kann bei Bedarf als XML-Datei heruntergeladen werden. Abbildung 8 zeigt, wie eine SignResponse Nachricht im Browser der BenutzerIn dargestellt wird.

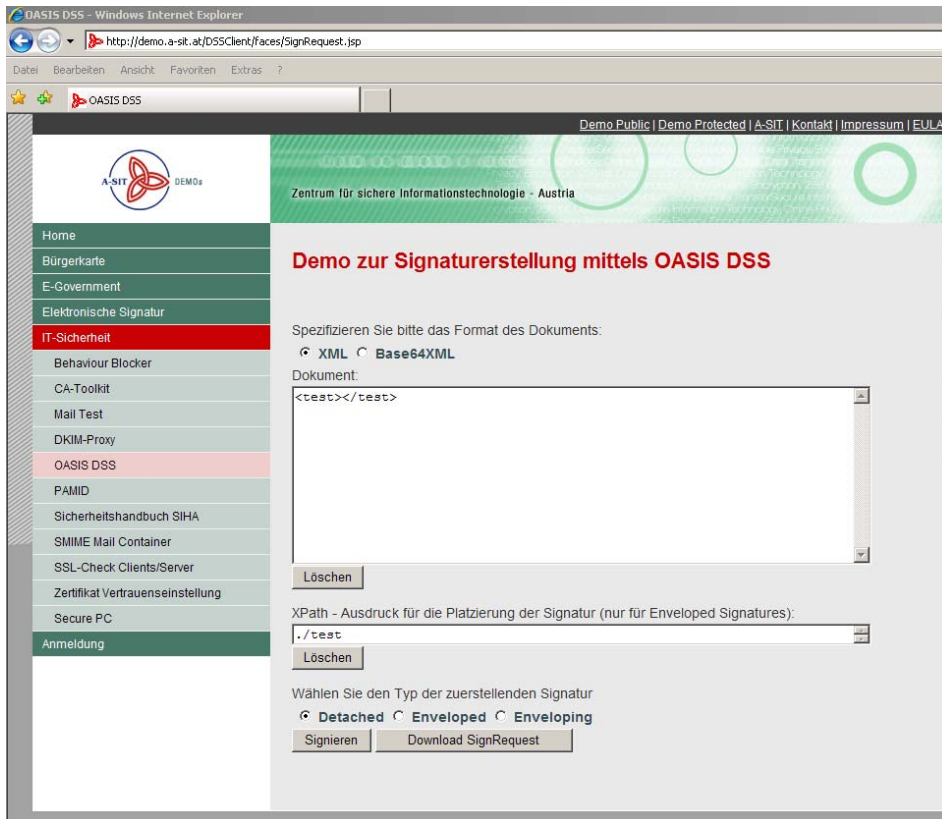


Abbildung 7 - Screenshot SignRequest

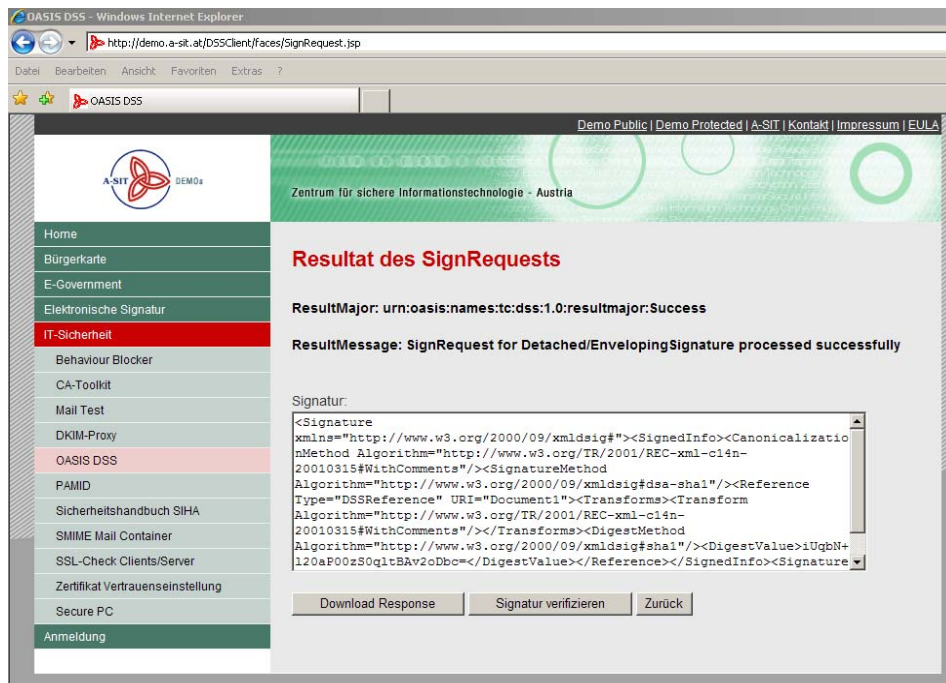


Abbildung 8 - Screenshot SignResponse

4.2 Signaturverifikation

Die mit dem Signaturerstellungs-Web-Interface erstellten Signaturen müssen natürlich auch verifiziert werden können. Dafür wird ebenfalls ein Web-Interface in der Demo-Installation zur Verfügung gestellt. Wie auch bei der Signaturerstellung können die folgenden Signatur-Typen verifiziert werden:

- Detached Signature
- Enveloped Signature
- Enveloping Signature

Wie im Fall einer Signaturerstellung wird die entsprechende Anfrage durch den DSS-Client erstellt und an den DSS-Server gesendet.

Abbildung 9 zeigt einen Screenshot des Web-Interfaces zur Erstellung eines DSS-VerifyRequests.

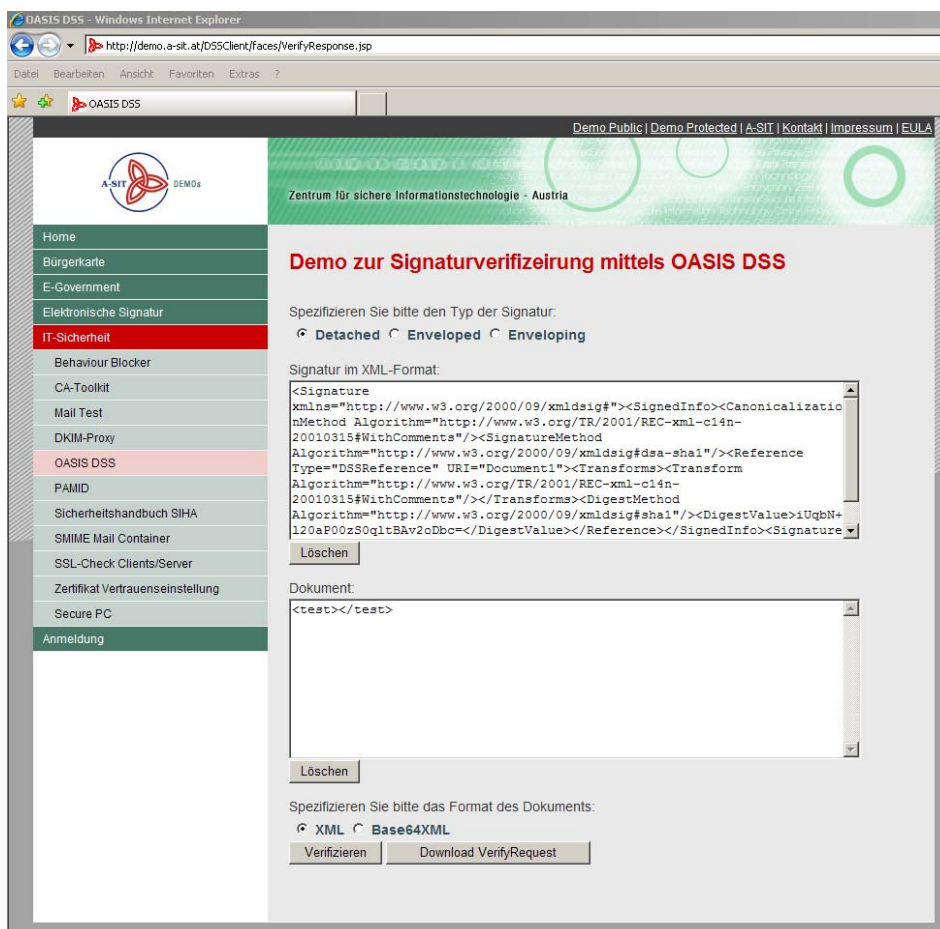


Abbildung 9 - Screenshot VerifyRequest

Abbildung 10 zeigt die Anzeige der aus dem DSS-VerifyRequest resultierenden DSS-VerifyResponse des DSS-Servers. An der jeweiligen Fehlermeldung des DSS-Servers kann man erkennen, ob die Signaturprüfung erfolgreich war oder fehlgeschlagen ist. Bei Bedarf kann die vom DSS-Server generierte Response auch heruntergeladen werden.

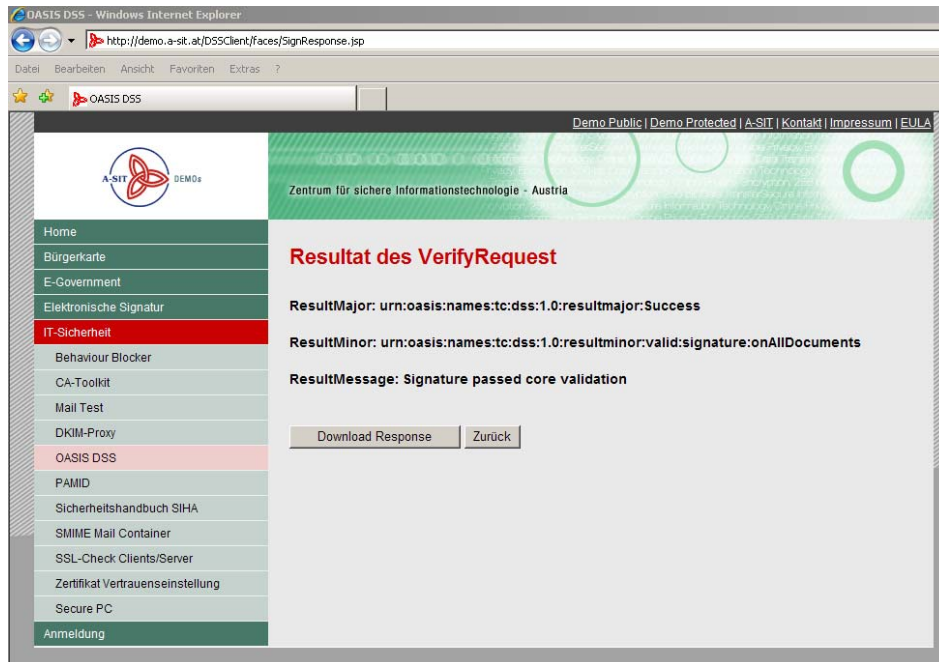


Abbildung 10 - Screenshot VerifyResponse

5 Zusammenfassung

Mit der ständig steigenden Relevanz von elektronischen Kommunikationsmedien steigt auch zunehmend die Bedeutung von elektronischen Signaturen. Da deren Erstellung bzw. Verifikation für Firmen eine gewisse organisatorische und sicherheitstechnische Herausforderung darstellt, bietet sich die Verwendung von Server-seitigen Lösungen an, um diesen Herausforderungen in geeignetem Maße zu begegnen. Mit OASIS-DSS steht seit kurzem ein Standard zur Verfügung, welcher die Verwendung einer Server-seitigen Anwendung zur Erstellung und Verifikation von elektronischen Signaturen spezifiziert.

Ziel dieses Projekts war es, diesen Standard zu analysieren, in Form einer Referenzimplementierung umzusetzen und über einen Demoserver zugänglich zu machen. Zu diesem Behufe sollte darüber hinaus ein DSS-Client implementiert werden, der eine einfache Verwendung der Demoinstallation ermöglicht.

Basierend auf diesen Projektzielen wurde der Standard OASIS-DSS zu Beginn des Projekts einer eingehenden Analyse unterzogen. Ziel dieser Analyse war es, die Möglichkeiten, welche dieser Standard AnwenderInnen bietet, sowie dessen Vorteile gegenüber herkömmlichen Verfahren zu ergründen und daraus eine für eine effiziente Implementierung geeignete Struktur zu finden.

Der Analysephase folgte eine auf den Erkenntnissen der Analyse aufbauende Referenzimplementierung des Standards, welche die Grundfunktionalität der Spezifikation umsetzt und so für AnwenderInnen bereitstellt. Die erstellte Implementierung ist als Web-Service ausgeführt, an welches Anfragen zur Erstellung und Verifikation von elektronischen Signaturen gesendet werden können und gliedert sich in drei Module. Das Modul DSS-Library implementiert dabei die für eine Verarbeitung von elektronischen Signaturen notwendige Funktionalität, während das Module DSS-Binding für das Mapping zwischen DSS-inherenten Nachrichten und gängigen Transportprotokollen zuständig ist und damit primär der Kommunikation zwischen der DSS-Library und dem aufrufenden Client dient. Die für eine flexible Konfiguration der Implementierung nötige Logik wird schließlich durch das Modul DSS-Configuration implementiert.

Nach Fertigstellung der Referenzimplementierung wurde gemäß der definierten Projektziele ein DSS-Client entwickelt, der es BenutzerInnen ermöglicht, auf die durch die Referenzimplementierung umgesetzte Funktionalität einfach zugreifen zu können. Dazu steht ein Web-Interface zur Verfügung, über das die BenutzerIn sehr einfach die zu verarbeitenden Daten an den DSS-Client und damit in weiterer Folge an das Web-Service übermitteln kann. Auf diese Weise können Signaturen über benutzerdefinierte Daten erstellt, bzw. bereits vorhandene Signaturen überprüft werden. Das Ergebnis der über das Web-Interface angestoßenen Verarbeitungsschritte wird der BenutzerIn wiederum über dieses Interface mitgeteilt.

Die während der Durchführung dieses Projekts gewonnenen Erkenntnisse und Erfahrungen konnten über eine OASIS-Mitgliedschaft in die Weiterentwicklung des Standards eingebracht werden.

Mit OASIS-DSS und der in diesem Projekt erstellten Implementierungen steht Unternehmen und interessierten AnwenderInnen eine Möglichkeit zur Verfügung, die Server-seitige Verarbeitung von elektronischen Signaturen in der Praxis auf einfach Art und Weise und ohne großen Aufwand kennenzulernen.

Glossar

Abkürzung	Bedeutung
BGBl	Bundesgesetzblatt
CMS	Cryptographic Message Syntax
DSS	Digital Signature Service
HTTP	Hyper Text Transfer Protocol
HTTPS	Hyper Text Transfer Protocol Secure
OASIS	Organization for the Advancement of Structured Information Standards
PKI	Public Key Infrastructure
SOAP	Simple Object Access Protocol
TLS	Transport Layer Security
URL	Uniform Resource Locator
XML	Extensible Markup Language

Referenzen

[Ref01]	Elektronische Signaturen: Werkzeuge : E-Government : Digitales Österreich [http://www.digitales.oesterreich.gv.at/site/5567/default.aspx]
[Ref02]	OASIS Digital Signature Service Core Protocols, Elements, and Bindings [http://docs.oasis-open.org/dss/v1.0/oasis-dss-core-spec-v1.0-os.html]
[Ref03]	Hypertext Transfer Protocol -- HTTP/1.1 [http://www.ietf.org/rfc/rfc2616.txt]
[Ref04]	SOAP specifications [http://www.w3.org/TR/soap/]
[Ref05]	SOAP 1.2 Attachment Feature [http://www.w3.org/TR/soap12-af]
[Ref06]	The TLS Protocol Version 1.0 [http://www.ietf.org/rfc/rfc2246.txt]
[Ref07]	Demo Server A-SIT [http://demo.a-sit.at]

Historie

Version 0.1	Datum 28.11.2007	Kommentar Erstellung der Dokumentstruktur
Ersteller Thomas Zefferer		
Version 0.2	Datum 30.11.2007	Kommentar Erstellen der Inhalte Teil 1
Ersteller Thomas Zefferer		
Version 0.3	Datum 30.11.2007	Kommentar Erstellen der Inhalte Teil 2
Ersteller Bernd Zwattendorfer		
Version 0.4	Datum 03.12.2007	Kommentar Überarbeitung
Ersteller Thomas Zefferer		
Version 0.5	Datum 03.12.2007	Kommentar Korrektur
Ersteller Herbert Leitold		
Version 1.0	Datum 04.12.2007	Kommentar Letzte Anpassungen
Ersteller Bernd Zwattendorfer		