

Dokumentation Signatur-Container

Signaturen im E-Government

Version 1.0, 19. Oktober 2006

DI Klaus Stranacher – kstranacher@iaik.tugraz.at

Zusammenfassung:

Im Zuge von E-Government Prozessen besteht immer wieder die Notwendigkeit elektronisch signierte Dokumente zu überprüfen. Enthalten diese Dokumente Referenzen auf andere Daten, so müssen diese zusätzlich mit dem signierten Dokument übermittelt werden. Um diesen Vorgang zu vereinfachen soll ein so genannter Signatur-Container geschaffen werden, der sämtliche Daten enthält, um eine Signaturprüfung zu ermöglichen.

Die vorliegende Dokumentation beschäftigt sich mit der Spezifikation eines solchen Signatur-Containers. Zuvor werden jedoch noch Anforderungen an den Container festgelegt und in Frage kommende Archivformate hinsichtlich ihrer Eignung untersucht.

Inhaltsverzeichnis:

Abbildungsverzeichnis.....	2
Revision History	3
1 Kurzbeschreibung	4
1.1 Grundlagen	4
1.2 Anforderungen	4
2 Archivformate	6
2.1 ZIP-Format	6
2.2 MIME-Format	7
2.3 Proprietäres Format	7
2.4 Folgerung	7
3 Spezifikation Signatur-Container.....	9
3.1 Aufbau und Struktur	9
3.2 Selbstextrahierung und Autorun	10
3.3 Prinzipieller Ablauf	10
4 Schlussfolgerungen.....	11
Referenzen.....	12

Anmerkung: Zur besseren Lesbarkeit wurde in diesem Dokument teilweise auf geschlechtsspezifische Formulierungen verzichtet. Die verwendeten Formulierungen richten sich jedoch ausdrücklich an beide Geschlechter.

Abbildungsverzeichnis

Abbildung 1.1: Prinzip Signatur-Container.	4
Abbildung 3.1: Aufbau und Struktur des Signatur-Containers.....	9
Abbildung 3.2: Prinzipieller Ablauf bei Verwendung des Signatur-Containers.	10

Revision History

Version	Datum	Autor(en)	
0.1	25.09.2006	Klaus Stranacher	Dokumenterstellung
0.2	02.10.2006	Klaus Stranacher	Erweiterungen
1.0	19.10.2006	Klaus Stranacher, Thomas Rössler	Überarbeitungen

1 Kurzbeschreibung

1.1 Grundlagen

Innerhalb des österreichischen E-Government ergibt sich die Anforderung elektronisch signierte Dokumente (Bescheide, Beglaubigungen, usw.) auf ihre Gültigkeit zu überprüfen. Diese Dokumente können potentiell weitere Daten wie beispielsweise Bilder oder Stylesheets enthalten. Um dem Empfänger eine erfolgreiche Überprüfung der Signatur zu ermöglichen, müssen ihm alle referenzierten Daten zur Verfügung stehen. Um die Übermittlung dieser Daten zu erleichtern, werden in diesem Dokument Möglichkeiten aufgezeigt, wie diese in einem so genannten *Signatur-Container* zusammengefasst werden können.

Abbildung 1.1 veranschaulicht das Prinzip eines solchen Signatur-Containers. Es zeigt ein signiertes Dokument, das neben zwei Bildern auch ein Stylesheet beinhaltet. Diese Daten werden in einem Signatur-Container zusammengefasst und anschließend an den Empfänger (Bürger oder Behörde) übermittelt. Auf Empfängerseite kann dann eine Überprüfung der Signatur des Dokuments erfolgen.

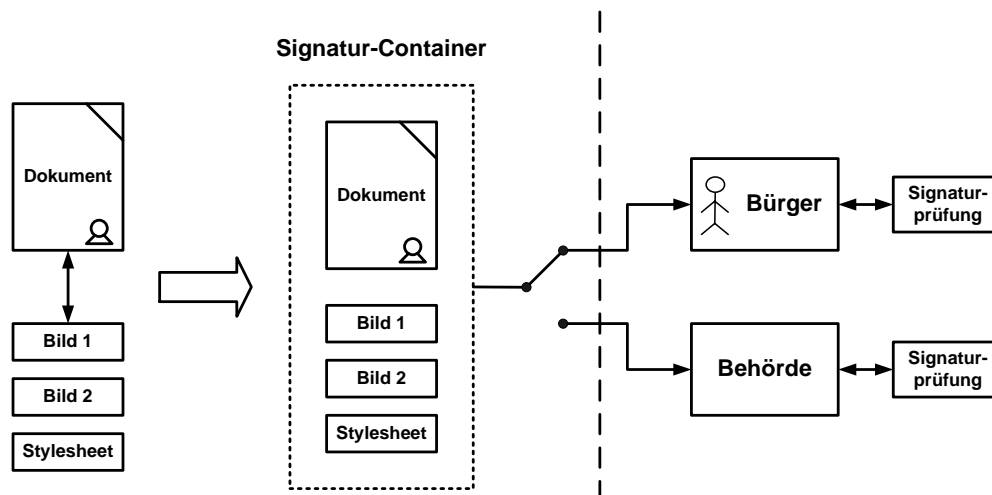


Abbildung 1.1: Prinzip Signatur-Container.

1.2 Anforderungen

Im Folgenden werden die Anforderungen an einen solchen Signatur-Container festgelegt:

- Adressierung/Halten aller Daten:
 - Der Signatur-Container muss sämtliche Daten enthalten, die für die Signaturprüfung notwendig sind. Hierzu gehören neben dem signierten Dokument auch alle Daten, die in diesem Dokument referenziert sind.
- Plattformunabhängigkeit:
 - Der Signatur-Container muss plattformunabhängig nutzbar sein.
- Selbstextrahierung mit Autorun:
 - Diese Daten, insbesondere das signierte Dokument, sollen auf einfache Art und Weise angezeigt werden können. Hierbei soll das Container-Format eine einfache Möglichkeit zur Selbstextrahierung und zum automatischen Aufruf bzw. zur automatischen Anzeige eines bestimmten Dokuments bieten.

- Es soll die Möglichkeit bestehen, einen Request zur Überprüfung der Signatur zu erstellen.
- Weiters muss ein Hinweis vorhanden sein, wie die Signatur geprüft werden kann.
- Unabhängigkeit:
 - Der Signatur-Container soll ohne spezielle Client-Software angesehen, ausgeführt und behandelt (geprüft) werden können.

2 Archivformate

Dieser Abschnitt befasst sich mit möglichen Archivformaten für den Signatur-Container. Hierbei kommen grundsätzlich folgende drei Formate in Frage:

- ZIP: Standard Format zur Archivierung von Dateien
- MIME: Nachrichtenformat zum Austausch von E-Mails und anderen Nachrichten.
- Proprietäres Format: Definition eines selbst entwickelten Archivformats

Im Folgenden werden diese drei Formate eingehender betrachtet und auf ihre Eignung als Signatur-Container Format eingegangen. Insbesondere werden diese Container-Formate den im vorangegangenen Abschnitt definierten Anforderungen gegenübergestellt. Diese lassen sich zusammenfassen als:

- (1) Adressierung der Daten,
- (2) Plattformunabhängigkeit,
- (3) Selbstextrahierung mit Autorun,
- (4) Abhängigkeiten

2.1 ZIP-Format

Dieses Format ist ein offenes Format zur Archivierung von Daten und ist in [ZIPSpec] spezifiziert. Es definiert einen Datencontainer, der beliebige Daten (komprimiert oder unkomprimiert) aufnehmen kann.

2.1.1 Adressierung der Daten

Die Spezifikation des ZIP-Formats sieht vor, dass zu jeder hinzugefügten Datei eine Pfadinformation angegeben werden kann. Diese Information kann sowohl absolute als auch relative Pfade enthalten. Bei der Verwendung als Format für den Signatur-Container kommt dabei nur eine relative Pfadangabe in Frage.

2.1.2 Plattformunabhängigkeit

Das ZIP-Format ist als offener Standard definiert und es existieren daher eine Reihe von Anwendungen, um ZIP-Archive zu erzeugen und zu entpacken. Es stehen hierbei Anwendungen für jede gebräuchliche Plattform zur Verfügung.

2.1.3 Selbstextrahierung und Autorun

Unter Selbstextrahierung wird verstanden, dass die Daten sich nach Öffnen des Archivs selbst entpacken. Mittels eines Autoruns soll parallel dazu ermöglicht werden eine bestimmte Datei zu öffnen und zur Anzeige zu bringen. Beide Funktionalitäten werden vom ZIP-Format unterstützt. Anwendungen die eine solche Selbstextrahierung mit Autorun unterstützen sind beispielsweise der kostenpflichtige *WinZip Self-Extractor* [WinZipSelf] oder die frei verfügbare Anwendung *ZIP 2 Secure EXE* [ZIP2Secure]. Anzumerken ist, dass ein solches Archiv nur unter Windows ausgeführt werden kann und somit die Funktion der Selbstextrahierung mit Autorun Windows-Benutzer vorbehalten bleibt. Die übrigen Archivfunktionalitäten bleiben jedoch unter den anderen Plattformen erhalten.

2.1.4 Abhängigkeiten

Um ein ZIP-Archiv anzeigen beziehungsweise entpacken zu können, benötigt man eine Anwendung, die in entsprechender Vielzahl zur Verfügung stehen. MS Windows hat dabei seit Windows XP das ZIP-Format direkt in das Betriebssystem integriert und es sind dort daher keine externen Programme nötig um mit ZIP-Archive umzugehen.

2.2 MIME-Format

MIME (Multipurpose Internet Mail Extensions) definiert ein Nachrichtenformat, das die Struktur von E-Mails und anderen Nachrichten festlegt und ist in [RFC2045] spezifiziert. Ähnlich wie im ZIP-Format können hier unterschiedliche Daten gemeinsam in einer Nachricht gekapselt und an den Empfänger übergeben werden.

2.2.1 Adressierung der Daten

Innerhalb einer MIME-Nachricht werden die Daten durch so genannte boundary strings getrennt; können auf diese Art, einer Verzeichnisstruktur vergleichbar, hierarchisch strukturiert werden.

2.2.2 Plattformunabhängigkeit

Das Nachrichtenformat MIME ist ein offener Standard, der nur den Aufbau und die Struktur einer Nachricht festlegt. Es gibt dadurch keine Plattformabhängigkeit.

2.2.3 Selbstextrahierung und Autorun

MIME bietet an und für sich keine Möglichkeit der Selbstextrahierung und eines Autoruns. Durch die Übermittlung via E-Mail wird die MIME-Nachricht aber im E-Mail-Client entsprechend der enthaltenen Dokumente und Dateien angezeigt. Die Funktion eines Autoruns ist jedoch nicht möglich.

2.2.4 Abhängigkeiten

Für die Anzeige einer MIME-Nachricht muss eine Viewer-Komponente vorhanden sein, wie zum Beispiel ein MIME-fähiger E-Mail-Client. Diese Bedingung kann jedoch heutzutage als breit und plattformunabhängig gegeben vorausgesetzt werden.

2.3 Proprietäres Format

Grundsätzlich besteht auch die Möglichkeit ein proprietäres Archivformat für den Signatur-Container zu definieren. Die österreichische E-Government Strategie sieht aber explizit den Einsatz offener und etablierte Standards vor, was im Widerspruch zur Definition eines eigenen Formates steht. Aus diesem Grund wird auf diese Möglichkeit nicht näher eingegangen.

2.4 Folgerung

Tabelle 2.1 stellt die grundlegenden Eigenschaften der in den vorherigen Abschnitten beschriebenen Archivformate zusammen. ZIP-Archive sind für den Großteil der Anwender ein gängiger Begriff. Die Benutzer sind dadurch im Umgang mit solchen Archiven geschult. Aus diesen Gründen und den angeführten Eigenschaften sehen wir das ZIP-Format als das geeignete Format für den Signatur-Container.

	ZIP-Format	MIME-Format
Adressierung der Daten	Absolut und relativ	Daten durch boundary strings getrennt
Plattformunabhängigkeit	Ja ¹	Ja
Selbstextrahierung	Möglich	via E-Mail Client
Autorun	Möglich	Nicht möglich
Abhängigkeiten	ZIP-Anwendung oder durch das Betriebssystem gegebene Unterstützung	E-Mail-Client erforderlich

Tabelle 2.1: Prinzip Signatur-Container.

Der folgende Abschnitt widmet sich nun der Spezifikation des Signatur-Containers.

¹ Selbstextrahierung mit Autorun nur unter Windows möglich.

3 Spezifikation Signatur-Container

Dieser Abschnitt beinhaltet die Spezifikation des Signatur-Containers. Das grundlegende Archivformat ist eine selbstextrahierende ZIP-Datei, deren Aufbau und Struktur im nun folgenden Abschnitt festgelegt wird.

3.1 Aufbau und Struktur

Der Signatur-Container muss sämtliche Daten enthalten um eine Signaturprüfung zu ermöglichen. Abbildung 3.1 zeigt den schematischen Aufbau und die Struktur des Containers.

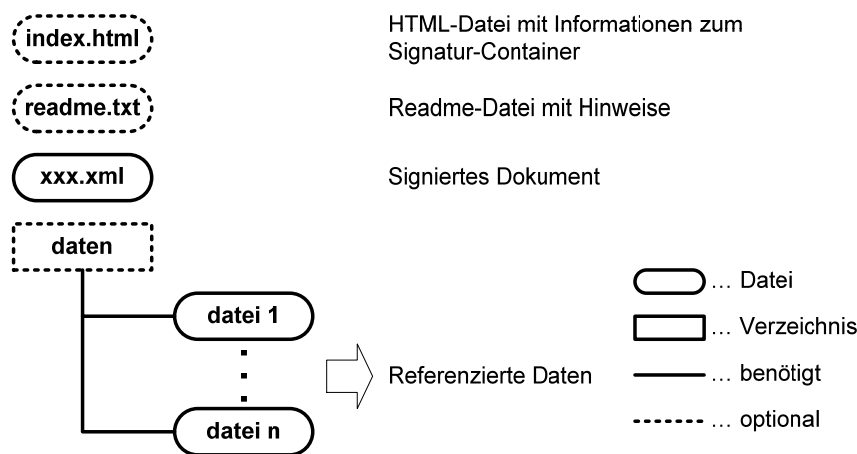


Abbildung 3.1: Aufbau und Struktur des Signatur-Containers.

Folgende konkreten Daten und Dokumente sollen im Signatur-Container enthalten sein:

- **index.html**

Diese Datei soll aufgerufen werden, wenn die Selbstextrahierung mit Autorun durchgeführt wird. Sie soll folgende Informationen und Möglichkeiten beinhalten:

- Allgemeine Informationen zum Container und Anzeige des Inhalts.
- Möglichkeit das signierte Dokument zu öffnen und anzusehen.
- Hinweise zur Signaturprüfung (wie beispielsweise das Hochladen auf ein entsprechendes Prüfservice) oder die Erstellung eines Signaturprüfrequest um das signierte Dokument mit der Bürgerkartenumgebung zu überprüfen.

Hinweis: Die Erstellung eines solchen Requests ist nicht Teil der Spezifikation und wird daher nicht eingehender behandelt. Die Spezifikation des Signatur-Containers sieht an dieser Stelle nur vor, dass eine Prüfrequest erzeugt werden kann.

Sollte eine `xxx.xml`-Datei bereits eine von selbst – bspw. via XSL-Transformationen – anzeigbare Datei sein, so kann anstelle der optionalen aber empfohlenen `index.html` die `xml`-Datei als Hauptdatei gesetzt und via Autorun dargestellt bzw. ausgeführt werden. Dies wäre zum Beispiel bei XML-Bescheiden in Verbindung mit einer für Browser interpretierbaren XSL-Transformations-Instruktion möglich. Bei anderen Fällen oder wenn dies grundsätzlich nicht möglich ist, dann muss eine `index.html`-Datei als Hauptdatei vorgesehen werden.

- **readme.txt**

Die `readme`-Datei dient dazu um allfällige weitere Informationen zu speichern, wie beispielsweise Informationen über den Container-Inhalt für Benutzer, die eine selbstextrahierende ZIP-Datei nicht ausführen können. Diese Datei ist optional.

- **xxx.xml**
 Diese Datei stellt das signierte Dokument dar. Der Name dieser Datei ist beliebig, er muss lediglich mit dem angegebenen Link in *index.html*, sofern die Hauptdatei *index.html* im Container vorgesehen ist, übereinstimmen. Wie in der Beschreibung zur *index.html*-Datei erwähnt, kann aber auch diese XML-Datei als Hauptdatei des Containers gesetzt werden, für Autorun, etc.
- **daten**
 Dieses Verzeichnis enthält sämtliche im signierten Dokument referenzierten Daten. Bei Dokumenten, die keine solchen Referenzen enthalten, kann dieses Verzeichnis entfallen. Bei der Benennung der referenzierten Dateien ist darauf zu achten, dass diese exakt mit den im Dokument angegebenen Referenznamen übereinstimmen

Bei der Erzeugung des Signatur-Containers ist weiters darauf zu achten, dass jede Datei mit relativen Pfadangaben hinzugefügt wird. Es dürfen also keinerlei absolute Pfade im Signatur-Container vorkommen.

3.2 Selbstextrahierung und Autorun

Für die Selbstextrahierung mit Autorun muss ein ZIP-Archiv entsprechend der oben definierten Struktur erzeugt werden. Anschließend muss mit einer entsprechenden Anwendung (beispielsweise [WinZipSelf] oder [ZIP2Secure]) eine selbstextrahierende ZIP-Datei erzeugt werden, die nach der Extrahierung die Datei *index.html* defaultmäßig öffnet.

3.3 Prinzipieller Ablauf

Abbildung 3.2 zeigt den prinzipiellen Ablauf bei Verwendung des Signatur-Containers nachdem dieser an den Benutzer übermittelt wurde. Der Benutzer führt den selbstextrahierenden Signatur-Container aus und er bekommt die Datei *index.html* angezeigt. Diese Datei beinhaltet den allgemeinen Informationen zum Container und der Anzeige des Container-Inhalts. Weiters enthält sie einen Link auf das signierte Dokument damit diese angezeigt werden kann. Neben Hinweisen zur Signaturprüfung kann diese Datei auch einen Link bzw. Prüfbutton beinhalten, der aus den vorhandenen Daten im Container einen Signaturprüfrequest generiert, der anschließend an die Bürgerkartenumgebung übermittelt werden kann.

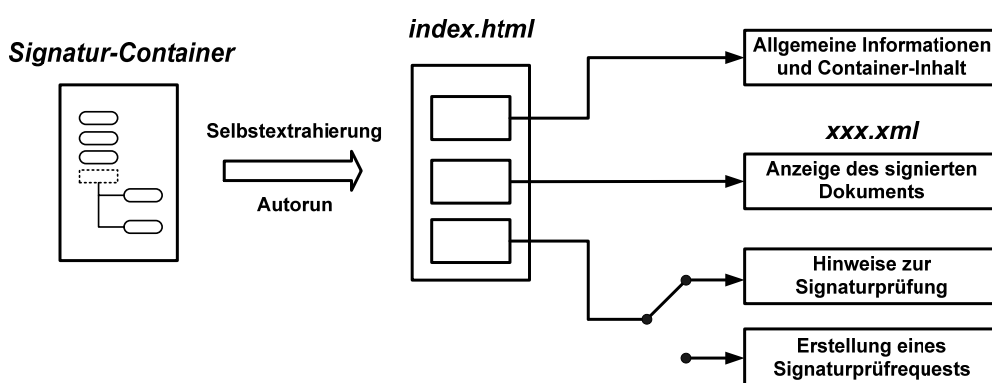


Abbildung 3.2: Prinzipieller Ablauf bei Verwendung des Signatur-Containers.

4 Schlussfolgerungen

Der Signatur-Container soll dazu dienen die Signaturprüfung zu erleichtern, indem sämtliche Daten, die für die Prüfung benötigt werden, in ihm enthalten sind. Sobald sich der Signatur-Container etabliert hat, können Erweiterungen innerhalb der Bürgerkartenumgebung und der Web-Service orientieren Signaturprüfung angedacht werden. So wäre eine mögliche Erweiterung, dass beide Komponenten einen Signatur-Container übernehmen können, danach die darin enthaltene(n) Signatur(en) überprüfen und abschließend das Prüfergebnis wie gewohnt zurück liefern.

Referenzen

[RFC2045]	Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies, http://www.ietf.org/rfc/rfc2045.txt
[WinZipSelf]	WinZip Self-Extractor 3.0; http://www.winzip.com/
[ZIP2Secure]	ZIP 2 Secure EXE; http://www.chilkatsoft.com/ChilkatSfx.asp
[ZIPSpec]	ZIP File Format Specification, http://www.pkware.com/business_and_developers/developer/appnote/