XML-Signatur Anwendungsprofile als Weg zur Lösung des Präsentationsproblems

Thomas Kunz, Ulrich Pordesch, Andreas U. Schmidt

Seit fast zwei Jahren existiert mit dem XML Signaturstandard [XML-DSig] ein Format für elektronische Signaturen, das weit über herkömmliche Signaturformate hinausgeht. Flexibilität und Mächtigkeit des Standards bergen aber auch erhebliche Gefahren. Dieser Beitrag untersucht, welche Risiken sich in Bezug auf das Präsentationsproblem bei der Anwendung von XML-Signaturen ergeben und welche Anforderungen sich daraus für die anwendungsspezifische Ausgestaltung und Einschränkungen des grundlegenden Standards ergeben.

[FOTO] Thomas Kunz
Wiss. Mitarbeiter des Instituts für Sichere Telekooperation (SIT) der Fraunhofer Gesellschaft, Darmstadt

E-Mail: Thomas.Kunz@sit.fhg.de

[FOTO] Dr. Andreas U. Schmidt

Wiss. Mitarbeiter des Instituts für Sichere Telekooperation (SIT) der Fraunhofer Gesellschaft, Darmstadt
E-Mail: Andreas.Schmidt@sit.fraunhofer.de

[FOTO] Dr. Ulrich Pordesch

Wiss. Mitarbeiter des Instituts für Sichere Telekooperation (SIT) der Fraunhofer Gesellschaft, Darmstadt
E-Mail: Pordesch@sit.fraunhofer.de

1. Einleitung

Durch digitale Signaturen ist es möglich, die Integrität und Urheberschaft von Daten zu überprüfen. Für eine hohe Beweiskraft elektronisch signierter Dokumente in Gerichtsprozessen ist dies allein jedoch nicht ausreichend: Es muss auch die Bedeutung der Daten anhand einer Präsentation nachgewiesen werden. Allerdings ist die Präsentation bei herkömmlichen Datenformaten wie z.B. Word häufig nicht eindeutig. Schon allein ein Ändern oder ein Nichtvorhandensein von unsignierten Formatvorlagen oder Zeichensätzen kann zu unterschiedlichen Darstellungen führen, die unterschiedliche Interpretationen durch den Anwender hervorrufen können. Auch die Interaktion des Anwenders mit seiner Software kann zu unterschiedlichen Präsentationen führen. Ein weiteres Problem liegt darin, dass neben sichtbaren auch diverse nicht-sichtbare Daten mitsigniert werden (zum Präsentationsproblem und seinen vielfältigen Ausprägungen und rechtlichen Folgen siehe [Pord_2003]).

Herkömmliche Nutzdatenformate wie Word weisen zahlreiche derartige Probleme auf. Hinzu kommt, dass Syntax und Semantik dieser Datenformate in der Regel nicht offen gelegt sind, so dass es nahezu unmöglich ist, die Korrektheit der Präsentation zu verifizieren, und festzustellen, inwieweit ausreichende Festlegungen zur Präsentation getroffen wurden.

Präsentationsprobleme gibt es auch bei den Formaten für Signaturen und ihre Bestandteile. So legen das gängige Signaturformat CMS [RFC 2630] sowie Standards für Zertifikate und diverse Signaturattribute nicht fest, ob und wie diese Daten zu präsentieren sind. Das Format der signierten Nutzdaten lässt sich in Form eines Signaturattributes zwar angeben Diese Angabe unterliegt aber keinerlei Einschränkungen hinsichtlich der zulässigen Nutzdatenformate und wird zudem meist nur unspezifisch verwendet (Format: „data“). Es wäre sogar möglich, in parallelen Signaturen zu denselben Daten unterschiedliche Formate anzugeben.

Gängige Daten- und Signaturformate bieten daher bislang keine ausreichenden Vorkehrungen zur Lösung des Präsentationsproblems. Daher stellt sich die Frage, inwieweit XML-Standards zur Lösung des Problems tauglich sind bzw. verwendet werden können. Im Folgenden werden zunächst grundlegende Anforderungen hinsichtlich des Präsentationsproblems dargelegt, bevor auf die speziellen Vor- und Nachteile von XML und XML-Signaturen eingegangen wird. Wir diskutieren danach Lösungsmöglichkeiten durch generische Standards und Anwendungsprofile.

2. Anforderungen

Wenn mit Signaturen nicht nur Daten gegen unbemerkte Verfälschung geschützt, sondern von Menschen rechtsverbindliche Erklärungen abgegeben werden sollen, muss das Präsentationsproblem gelöst werden. Dabei sind sowohl Falschpräsentation zu vermeiden, wie auch die Mehrdeutigkeit von Daten hinsichtlich ihrer Präsentation. Vorkehrungen gegen das Falschpräsentationsproblem erfordern Sicherheitsvorkehrungen in Signaturanwendungen und im Betriebssystem, die weitgehend unabhängig vom Datenformat sind. Wir konzentrieren uns hier auf die vom Datenformat abhängigen Teile des Präsentationsproblems, welche insbesondere das Mehrdeutigkeitsproblem betreffen.

Generell ist es zunächst wichtig festzustellen, dass es, um Rechtsverbindlichkeit herzustellen, zumeist nicht notwendig ist, dass Daten nur auf eine einzige Art und Weise präsentiert werden können. Diese Zielvorgaben wären technisch praktisch auch kaum zu erreichen, da Daten – etwa

durch die Wahl unterschiedlicher Anwendungsprogramme immer unterschiedlich präsentiert werden können. Ebenso unnötig wie technisch kaum realisierbar ist es, nachzuweisen, ob und wie ein Signierer die Daten tatsächlich präsentiert hat. Zwar gibt es einzelne Rechtsbereiche, in denen die detaillierte Kenntnisnahme des Inhalts einer Erklärung gefordert und nachzuweisen ist (Extremfall: Testamentserklärungen), aber in diesen Bereichen ist die Abgabe von Erklärungen in elektronischer Form bislang aus gutem Grund unzulässig. In der großen Mehrheit möglicher Streitfälle vor Gericht, insbesondere vor Zivilgerichten, ist vielmehr rechtlich bedeutsam, dass eine elektronische Erklärung einem Erklärenden rechtlich *zugerechnet* werden kann. Dies bedeutet, dass ein Richter davon ausgehen darf, dass bestimmte signierte Daten eine bestimmte Erklärung repräsentieren, die der Signierer abgegeben hat. Der Richter muss sich dann nicht darum kümmern, ob und wie er die Daten selbst zuvor präsentiert hat. Allerdings ist diese Zurechnungsregel nur unter der Voraussetzung anwendbar, dass der Signierer zumindest die Möglichkeit hatte, festzustellen welche Erklärung er abgab und die Abgabe nicht gewünschter Erklärungen zu vermeiden. Aus dieser zwingenden Voraussetzung ergeben sich zahlreiche Anforderungen an die Signaturanwendungssysteme, wie auch die verwendeten Datenformate, von denen im Folgenden besonders wichtige genannt seien [Pord_2003].

Eindeutigkeit

Es muss klar sein, welche Präsentation den zu signierenden bzw. signierten Daten zugeordnet ist. Substantielle Abweichungen in den Präsentationen derselben signierten bzw. zu signierenden Daten sollten ausgeschlossen werden können. Die Eindeutigkeitsforderung betrifft das „ob“ und „wie“ der Präsentation der Daten. Es muss festgelegt sein, welche Bestandteile präsentiert werden sollen (relevanter Erklärungsinhalt, z.B. Formularinhalt) und welche nicht (Zusatzdaten, z.B. automatisch auszuwertende Formatvorlage). Für die zu präsentierenden Daten muss die Darstellung, etwa Layout auf dem Computerbildschirm, ebenso wie die vorzunehmende Interaktion, etwa Blättern von vorne nach hinten bei einer seitenorientierten Ausgabe, festgelegt sein. Wie genau die jeweiligen Festlegungen sein müssen, hängt vom jeweiligen Anwendungskontext ab. Allgemein lässt sich nur sagen, dass umso weniger mit Rechtsstreit um den Inhalt von elektronischen Erklärungen zu rechnen ist, je genauer die Festlegungen zur Präsentation der signierten Daten sind.

Transparenz

Der Signierer muss erkennen können, was er signiert, der Verifizierer muss erkennen können, was signiert wurde. Die Präsentation muss vollständig sein. Die obengenannte eindeutige Präsentation müssen Signierer und Verifizierer nicht zwangsweise durchführen. Eine andere Präsentation mag für die jeweiligen Zwecke geeigneter sein, als eine gerichtlich im Zweifelsfall zuzurechnende. Signierer und Verifizierer müssen die zuzurechnende Präsentation aber durchführen *können* und dürfen daran nicht gehindert werden. Dies setzt zum einen sichere oder zumindest nutzerkontrollierte Signiersysteme voraus, was hier nicht näher betrachtet wird. Zum anderen – und das ist hier relevant – muss das ob und wie der durchzuführenden Präsentation erkennbar sein, wenn der Signierer signiert und der Verifizierer verifiziert. Neben den zu signierenden Inhaltsdaten müssen daher auch zusätzliche Daten vorhanden sein, die dieses „ob“ und „wie“ beschreiben. Dazu gehören das Datenformat, damit zusammenhängend die zu nutzende Präsentationskomponente (etwa bestimmter Viewer), die konkrete Darstellung der Inhalte durch diesen Viewer (etwa dargestellt durch ein XSL-Stylesheet) und die Interaktion des Nutzers mit diesem Viewer.

Eine weitere Transparenzanforderung ergibt sich durch die Möglichkeit, zu signierende Dokumente vor der Erzeugung der Signatur zu transformieren und Teile zu selektieren. Signierer und Verifizierer sollten klar erkennen können, welche Teile des Ausgangsdokumentes signiert und damit gegen Verfälschung gesichert sind und welche nicht.

Sicherheit

Voraussetzung für die Zurechenbarkeit ist, dass Signierer und Verifizierer die Möglichkeit haben müssen, zu signierende bzw. signierte Dokumente korrekt zu präsentieren und Falschpräsentationen zu vermeiden. Das betrifft die Sicherheit des Signiersystems und seiner Komponenten gegenüber Fehlern und Manipulationen (was hier nicht betrachtet wird). Es hat auch den Aspekt, dass Benutzerfehler bei der Interaktion vermieden werden sollen. Dieser Aspekt ist auch von den signierten Daten und deren Format abhängig. Ist die Präsentation zwar theoretisch eindeutig festgelegt, erfordert aber komplexe Interaktion (erst Dokumenteigenschaften ansehen, Kommentare ignorieren, bestimmte Bildschirmeinstellung wählen, verborgenen Text einblenden etc.), dann sind Nutzerfehler regelrecht „vorprogrammiert“ und die rechtliche Zurechenbarkeit kann schon deshalb in Frage stehen. Daraus ergibt sich die Anforderung, dass die signierten Daten „einfach“ zu präsentieren sind, d.h. keine komplexe und fehleranfällige Interaktion voraussetzen. Einfach präsentierbar sind Daten, die statisch sind (keine aktiven Inhalte aufweisen), linear aufgebaut und am besten auch ausdruckbar sind.

Eine weitere Sicherheitsanforderung, die wiederum mit der Möglichkeit beliebiger Transformation zusammenhängt, ist folgende: Ein zu schützendes Ausgangsdokument soll nur so transformiert werden, dass Verfälschungen durch Verifikation der Signaturen des Zieldokumentes erkennbar werden.

Beweiseignung

Über die genannten Eigenschaften müssen Beweise vor Gericht geführt werden, u.U. nach längerer Zeit. Richter müssen, notfalls unter Hinzuziehung von Sachverständigen, überzeugt werden. Dies ist schwierig, wenn Präsentationsregeln (Formatspezifikationen, Layoutbeschreibungen, Interaktionsregeln) nicht offengelegt oder nur in flüchtiger, veränderbarer Form hinterlegt sind. Die genannten Regeln sollten Teil der signierten Daten selbst werden und damit durch die Signatur umfasst und verifizierbar sein. Wenn Daten, die nicht durch den Nutzer verändert werden, sondern vorgegeben werden (etwa Layoutbeschreibungen eines vielverwendeten Formulars) aus Aufwandsgründen nicht mitsigniert werden, so müssen sie zumindest bei einer kontrollierten Instanz hinterlegt werden. Es reicht nicht aus, einen Link auf Daten zu signieren, die sich auf irgendeinem Rechner unter Verwaltung eines beliebigen, keiner Kontrolle unterliegenden Betreibers befinden.

3. XML als Signatur- und Datenformat

Ein für die Ausgabe in einem Browser bestimmtes XML-Dokument besteht generischerweise aus drei Teilen:

- den Dokumentdaten, d.h. den Inhalten und diese beschreibenden Metadaten (Tags),
- einem Verweis auf ein Schema, in dem die logische Struktur einer Dokumentklasse beschrieben wird und dem die Syntax der Dokumentdaten entsprechen soll,
- einem Verweis auf ein Stylesheet [Stylesheet_Ass], der beschreibt, wie die Dokumentdaten (entsprechend der darin angegebenen Metadaten) zu präsentieren sind.

Für Stylesheets wurde die XSL-Transformation Spezifikation [XSL-T] entwickelt. Diese erlaubt es, die XML-Daten in ein beliebiges präsentierbares Format (wie etwa HTML) umzuwandeln. Mögliches Zielformat ist z.B. die XSL-Formatting Language [XSL-FO], eine seitenorientierte Formatierungssprache für die Druck- oder Bildschirmausgabe.

XML-DSig

Ein XML-basiertes Format für Signaturen ist unter der Bezeichnung „XML-Signature Syntax and Processing“ entwickelt worden [XML-DSig]. XML-Signaturen sind flexibler und mächtiger als z.B. CMS-Signaturen:

- Signieren vieler Dokumente: Es können nicht nur ein, sondern beliebig viele Datenobjekte (XML-Dokumente oder auch andere Dokumente), signiert werden.
- Referenzierung: Die Datenobjekte müssen nicht lokal vorhanden sein, sondern können sich im Netz befinden, weil über Links (URI) auf sie verwiesen wird.
- Datenselektion/-transformation: Von jedem referenzierten Datenobjekt können [XPath, XPath-Filter] Teile (z.B. Formulardatenelemente eines XML-Formulars) gezielt zum Signieren ausgewählt werden. Mit XSL-Transformationen können die Datenobjekte bzw. die daraus selektierten Daten nahezu beliebig verändert werden, bevor signiert wird.

Der Vorgang der Generierung einer XML-Signatur wird durch den Standard so festgelegt: Zunächst sind die zu signierenden Dokumente auszuwählen und eventuell Teile zu selektieren und zu transformieren. Danach wird für jedes transformierte Objekt ein Hashwert berechnet. Aus dem Hashwert und weiteren Daten (wie die Verweisadressen auf die Datenobjekte) wird ein Signaturelement gebildet, das gehasht und anschließend mit dem Signaturschlüssel verschlüsselt wird.

Vorteile

XML weist gegenüber herkömmlichen Nutzdaten- und Signaturformaten einige grundlegende Vorteile auf. Zu nennen ist zunächst die Lesbarkeit der Kodierung, die die Interpretation und Analyse der Daten erleichtern kann. Der Hauptvorteil von XML hinsichtlich des Präsentationsproblems liegt in der klaren Trennung von Daten und Kontextkomponenten (Schemata, Layouts bzw. Transformationen) und der Beschreibung dieser Komponenten als menschenlesbarer Text. Dies erzwingt eine vollständige Offenlegung der Syntax, Präsentation und z.T. auch der Semantik, die damit einer Überprüfung zugänglich werden - z.B. dahingehend, ob alle Daten präsentiert werden. Kontextkomponenten können auch für Teile eines Dokuments definiert und zugewiesen werden. So können Syntax und Präsentation von Signaturen und Nutzdaten getrennt beschrieben werden, und es wird beispielsweise möglich, für die Darstellung einer Signatur auf eine globale Festlegung zu verweisen und nur die Präsentation der Nutzdaten dokumentspezifisch festzulegen [Schm_2000]. Verantwortlichkeiten für Teile der Syntax, Semantik und Präsentation eines Dokumentes können so getrennt werden. Durch die klare Trennung von Daten, Schema und Stylesheet können für dieselben Daten eines Schemas zudem verschiedene geeignete Stylesheets definiert werden, um den Bedürfnissen von Anwendern oder den Spezifika von Endgeräten Rechnung zu tragen.

Auch XML-DSig weist gegenüber herkömmlichen Signaturformaten Vorteile auf. Durch die Möglichkeit, mehrere Datenobjekte zu signieren, können Kontextkomponenten wie insbesondere assoziierte Schemata oder Stylesheets mitsigniert werden. Dabei kann auf in globaler Verantwortung stehende Komponenten, die im Netz zentral zugänglich gemacht werden können, leicht verwiesen werden. Des Weiteren können mittels Selektionen und Transformationen Teile eines Ausgangsdokumentes, die signiert werden sollen gezielt ausgewählt und andere unerwünschte Teile gezielt ausgeblendet werden. Dabei sind die verwendeten XSLT-Stylesheets bzw. XPath-Ausdrücke prinzipiell nachvollziehbar und es ist dadurch prinzipiell überprüfbar, welche Auswirkungen die Operationen auf die Daten, die signiert werden, haben.

Probleme

Wie gezeigt, weist das Signaturformat nach dem XML-Signaturstandard ein hohes Maß an Flexibilität auf, das aber zugleich hinsichtlich des Präsentationsproblems Risiken birgt.

Allein die Verwendung von XSL-Stylesheets bietet selbstverständlich keine Gewähr für die Vermeidung von Präsentationsproblemen ist. Denn deren Ziel kann jedes hinsichtlich der Präsentation mehrdeutige Format, wie etwa HTML, sein. Auch ist die Angabe von Kontextkomponenten, wie Stylesheets und Schemata und deren Formatierung als lesbarer Text keine Gewähr für die Verständlichkeit und die faktische Möglichkeit einer Überprüfung. XML-Dokumente mit komplexen Schemata mögen lesbar im Sinne von buchstabierbar sein, nicht unbedingt aber verständlicher als die Hex-Code-Darstellung einer binärkodierten Dokumentvorlage in Word. Verdeckte Inhalte, die mitsigniert werden, sind deshalb ebenso möglich, wie bei herkömmlichen Formaten.

Werden XML-Dokumente signiert, nicht aber die zugehörigen Stylesheets, so kann die Präsentation nachträglich verfälscht sein. Durch die Möglichkeit, beliebig viele Datenobjekte zu signieren, bietet XML-DSig anders als CMS zwar die Möglichkeit, solche Kontextkomponenten mitzusignieren. Hierbei treten dann aber weitere Probleme auf: Mitsignierte Komponenten sind nicht notwendigerweise diejenigen, die im XML-Dokument selbst referenziert sind. Die Kontextkomponenten können für die Interpretation einer Erklärung wichtige Angaben und zudem verdeckte Daten enthalten und müssten dann gegebenenfalls zusätzlich präsentiert werden. Der Standard legt auch nicht fest, wie eine auf die Präsentation bezogene Gültigkeitsprüfung der Signatur zu erfolgen hätte. Eine Gültigkeitsprüfung etwa dahingehend, ob referenzierte Stylesheets und Schemata und andere Objekte Teil der Signatur sein müssen, sieht XML-DSig nicht vor. Es gibt auch keine Regelung, dass ein mitsigniertes Stylesheet und ein mitsigniertes Schema mit darin eventuell enthal-

tenen wichtigen Informationen nicht zu präsentieren, sondern lediglich auf die XML-Daten anzuwenden sind.

Schließlich liegt in den vielfältigen Möglichkeiten, die zu signierenden Daten auszuwählen und zu verändern, ein weiteres Problem. Werden Selektionen und Transformationen unsachgemäß verwendet, so schützt die Signatur das Ausgangsdokument nicht mehr gegen Veränderungen. Im Extremfall haben die signierten Daten mit den ursprünglich referenzierten Datenobjekten überhaupt nichts mehr zu tun.

Im XML-Signaturstandard ist bislang auch nicht vorgesehen, der Signatur selbst eine Darstellung, etwa über ein Stylesheet, zuzuweisen. Zwar kann in einem XML-Dokument, das die Signatur enthält, ein XSL-Dokument angegeben werden. Diese Referenz befindet sich jedoch außerhalb der Signatur und ist damit nur dann signiert, wenn eine Referenz in der Signatur auf das Dokument verweist, in dem sich die Signatur befindet. In den anderen Fällen ist anhand eines signierten XML-Dokumentes nicht nachvollziehbar, wie die Signatur und ihre Bestandteile (Zertifikate usw.) zu präsentieren sind.

Die große Variabilität von XML-Signaturen eröffnet also einerseits interessante Anwendungsmöglichkeiten, birgt aber zugleich ein erhebliches Risiko fehlerhafter oder mehrdeutiger signierter Dokumente. Will man die Präsentationsprobleme mit XML vermeiden, so sind Einschränkungen dieser Variabilität oder zusätzliche Dokumentenstandards notwendig.

4. Lösungsansätze

Im Folgenden diskutieren wir, inwieweit den Anforderungen durch Einschränkungen oder Erweiterungen des XML-Signaturstandards oder andere Maßnahmen entsprochen werden kann.

Eingeschränkte Verwendung

Eine nahe liegende Möglichkeit ist es, zu versuchen, den grundlegenden XML-Signatur-Standard so zu verwenden, dass Präsentationsprobleme vermieden bzw. zumindest reduziert werden. Dazu wären mehrere Maßnahmen erforderlich: Für die Präsentation der transformierten Datenobjekte, die signiert werden, müssen evaluierte Viewer eingesetzt werden. Mindestanforderung an solche Viewer ist, dass sie die Daten hinsichtlich ihres Formates analysieren und eindeutig präsentieren. Ist eine eindeutige Präsentation nicht möglich, muss der Viewer die Mehrdeutigkeit in einer für den Signierer/Verifizierer leicht verständlichen Weise verdeutlichen oder den Vorgang mit einer Fehlermeldung abbrechen. Um Unklarheiten darüber zu vermeiden, ob und wie Stylesheets oder Schemata zu präsentieren sind, ist auf deren Mitsignierung zu verzichten.

Diese (und weitere zu diskutierende) Einschränkungen in der Verwendung des Standards sind allerdings gravierend. Sie machen Vorteile, die in der Flexibilität des Standards liegen, zunichte, und verhindern die Anwendbarkeit in komplexen Anwendungen, wie Workflows (z.B. Geschäftsprozesse und darin verwendete Formulare). Einige Probleme werden auch nicht richtig gelöst. So wird das Format der signierten Daten nicht explizit angegeben, sondern nur analysiert, woraus sich zumindest Zweifel ergeben können, ob dieses analysierte Datenformat zurechenbar ist. Nicht sicherzustellen ist zudem durch diesen Lösungsansatz, dass ein sicherer und zugleich transparenter Zusammenhang zwischen referenzierten Datenobjekten und dem signierten Ergebnis der Transformation besteht.

Alleine mit der Begrenzungen in der Verwendung des XML-Signaturstandards kommt man daher nicht sehr weit. Deshalb diskutieren wir im Folgenden, inwieweit den Anforderungen durch Einschränkungen oder Erweiterungen des XML-Signaturstandards oder weitergehende Maßnahmen entsprochen werden kann. Zwei Leitgedanken stehen hierbei im Vordergrund:

1. Evaluierung von Kontextkomponenten dahingehend, ob sie erforderliche Eigenschaften, wie insbesondere eine eindeutige Präsentierbarkeit, aufweisen
2. Verifizierbare Bindung evaluierter Kontextkomponenten an ein signiertes XML-Dokument, das heißt, aller verwendeten Ressourcen, wie Syntax- und Darstellungsbeschreibungen und weiterer Komponenten, zum Beispiel textuelle Beschreibungen und Metadaten.

Evaluierung von Kontextkomponenten

Kontextkomponenten, wie Schemata und Stylesheet-Transformationen, werden bei XML zwar prinzipiell offengelegt, sind aber zumindest durch Laien, die Dokumente signieren oder verifizieren, nicht direkt überprüfbar. Eine Sachverständigenprüfung im Einzelfall ist zwar prinzipiell möglich, aber aufwändig und wenn sie erst bei einem Gerichtprozess stattfindet, zu spät. Deshalb ist es nötig, Kontextkomponenten vor ihrer Verwendung durch unabhängige sachverständige Institutionen zu evaluieren, um die Beweiseignung zu erhöhen.

Gegenstand der erforderlichen Evaluierung sind zum einen XSL-Transformationen, durch deren Anwendung Datenobjekte, in zu signierende Datenobjekte transformiert werden. Hierbei ist zu berücksichtigen, dass mehrere Transformationen in Folge durchgeführt werden können. Die letzte Transformation erzeugt dabei diejenigen Daten, die einem evaluierten Viewer zuzuführen und eindeutig zu präsentieren sind. Um sicherzustellen, dass die Daten tatsächlich eindeutig präsentiert werden, könnte man diese letzte Transformation dahingehend evaluieren. Es wäre zu untersuchen, ob sie alle möglichen Eingaben in eindeutig präsentierbare Daten umsetzt, die dann durch einen Viewer problemlos dargestellt werden können. Die Eindeutigkeitsforderung wäre damit zu erfüllen, die Evaluierung würde zudem die Forderung nach Beweiseignung des Dokumentes sicherstellen.

Allerdings ist alleine mit dieser Maßnahme noch nicht sicherzustellen, dass das referenzierte Ausgangsdokument und das Ergebnis der Transformationen in einem sinnvollen Zusammenhang stehen. Das signierte Datenobjekt mag als Erklärung rechtlich zurechenbar sein, das referenzierte Ausgangs-Datenobjekt aber gegen Veränderungen dennoch ungeschützt sein. Will man einen weitergehenden Schutz erreichen, dann muss man Transformationen und Transformationsfolgen in Bezug auf Klassen von Ausgangsdokumenten evaluieren. Das bedeutet, dass durch eine Evaluierungsinstanz geprüft wird, ob XSL-Transformationen geeignet sind, ein Ausgangsdokument in ein Zieldokument unter Beachtung rechtlicher und weiterer Anforderungen zu überführen. Durch einfache Prüfung, ob solche Transformationen eingesetzt werden, ist dann schnell und ggf. automatisch entscheidbar, ob rechtsgemäß und sicher signiert wurde.

Neben Stylesheets sind Schemata zu evaluieren. Diese sind für die Eindeutigkeit der Präsentation zwar von geringer Bedeutung, können jedoch für die Interpretation der Erklärung relevante Inhalte aufweisen

(etwa Kommentare), die nicht zu Missverständnissen und Unklarheiten in einem späteren Gerichtsprozess führen dürfen. Zudem sind sie notwendigerweise Grundlage für die Evaluierung von darauf bezogenen Stylesheet-Transformationen.

Eine Evaluierung alleine reicht aber nicht aus. Evaluierte Kontextkomponenten müssen in beweiskräftiger Form an XML-Daten gebunden werden. Zusätzlich müssen Regeln angegeben werden, welches Datenformat ein transformiertes Datenobjekt hat und welche Präsentationskomponente zu starten ist, welche referenzierten Datenobjekte zu präsentieren sind und gegebenenfalls in welcher Reihenfolge und wie eine erweiterte Gültigkeitsprüfung der Signatur durchzuführen ist (z.B. ob Stylesheet und Schema von Ausgangs- oder Zieldokumenten mitsigniert sind).
Kontextbindung und die Angabe von zusätzlichen Regeln können im wesentlichen auf zwei Arten realisiert werden: Durch eine Erweiterung des XML-DSig-Standards oder die Schaffung neuer Standards für signierbare Dokumente, die auf XML-DSig aufbauen.

Erweiterungen des Standards

Eine Möglichkeit der Kontextbindung ist eine generische Erweiterung des XML-DSig Standards, in der die notwendigen Festlegungen mit Hilfe vorgegebener Datenstrukturen getroffen werden können.

Ein Beispiel für einen Ansatz in dieser Richtung ist XAdES [ETSI_2002]. Diese Spezifikation führt Signaturattribute ein, über die unter anderem ein Datenformat für jedes zu signierende Datenobjekt als MIME-Type oder ASN.1 Objekt-Bezeichner angegeben werden kann. Zusätzliche Festlegungen, wie beispielsweise, welche Datenobjekte zu präsentieren, welche Kontextkomponenten zu signieren und welche Stylesheets für Signaturen zu verwenden sind, sind zwar nicht vorgesehen, könnten aber theoretisch in analoger Weise realisiert werden.
Allerdings sind solche Erweiterungen kritisch: Sie überfrachten den Standard mit Details für die Präsentation, die sich anwendungsspezifisch stark unterscheiden werden. Von den eben genannten Problemen ist jede generische Erweiterung des XML-DSig Standards betroffen. Spezielle Erweiterungen für bestimmte Anwendungen und zur Lösung spezifischer Probleme scheinen aber aus technischen wie aus grundsätzlichen Erwägungen wenig erstrebenswert: Hier würden Misch-Standards entstehen, in denen weit über die eigentlichen Signaturen hinausgehende Beschreibungen dargestellt würden, z. B. anwendungsspezifische Einschränkungen der erlaubten Transformationen. Ein solches Vorgehen widerspräche der modularen Natur der W3C-XML Standards.

Formate für signierbare Dokumente

Will man den XML-Signaturstandard mit Aspekten der Kontextbindung, wie oben argumentiert, nicht überfrachten, so liegt es nahe, neue, möglichst generische Formate zu spezifizieren, die auf XML-DSig aufbauen bzw. diesen Standard verwenden. Ein generischer Überstandard für „signierbares XML-Dokument“ sollte ein (XML-) Datenformat beschreiben, in dem sich die Struktur von signierten XML-Dokumenten, ihre Darstellung, die Bindung der Kontextkomponenten, einschränkende Bedingungen an die (XML-) Signaturen (z. B. erlaubte Transformationen), etc. darstellen lassen. Eine Instanz eines solchen Überstandards wäre dann ein *XML-Signatur Anwendungsprofil*. In einem entsprechenden XML-Format für eine Art „signierbare Dokumente“ könnte beispielsweise festgelegt werden, dass ein zu signierendes Dokument bestehen muss aus:

- einem XML- Dokument, das die Dokumentdaten enthält,
- einem Schema-Dokument für dieses XML-Dokument,
- einem zertifizierten Stylesheet-Dokument, das zur Präsentation dieses XML-Dokumentes anzuwenden ist,
- einem zertifizierten Stylesheet-Dokument, das zur Präsentation der Signatur anzuwenden ist,
- weiteren Daten, etwa Einschränkungen erlaubter XML-Signaturen, Transformationen, etc.

Dies wäre zu ergänzen um weitere Angaben, wie das Datenformat und gegebenenfalls anwendungsspezifische Daten, wie der Anwendungsbereich und die Nutzergruppe. Bei der Signaturverifikation kann dann eine evaluierte Komponente für diese signierbaren Dokumente erweiterte Gültigkeitsprüfungen durchführen und passende, ebenfalls evaluierte Präsentationskomponenten starten, um das Dokument zu präsentieren.

Infrastrukturen

Um dem angestrebten Standard „signierbares Dokument“ für die praktische Anwendung „Leben einzuhauchen“, also seine Verwendung insbesondere bei den potenziellen Anbietern von Anwendungsprofilen attraktiv zu machen, scheint die Errichtung einer stützenden Infrastruktur ratsam. Vorstellbar ist die Einrichtung öffentlicher **Re-**

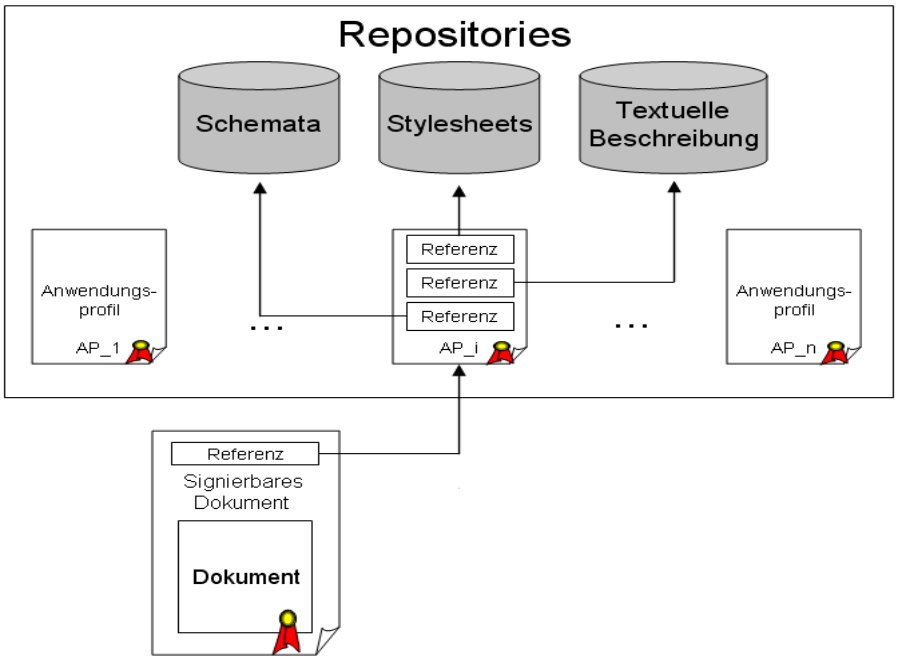

Abbildung 1: Grobstruktur für eine Anwendungsprofil-Infrastruktur.

positories mit geeigneten und gegebenenfalls evaluierten und zertifizierten Ressourcen: Schemata, Stylesheets zur Darstellung von Signaturen, Zertifikaten und Dokumenten, sowie die Anwendungsprofile für bestimmte Zwecke selbst – dargestellt als XML-Dokumente innerhalb des angesprochenen Standards „signierbares Dokument“. Diese speziellen Anwendungsprofile können die Ressourcen dann in ihren Kontext einbinden. Die Repositories müssen geeignete Sicherheitsmerkmale aufweisen, zum Beispiel ist die Integrität der Ressourcen durch geeignete Maßnahmen wie Issuer-Zertifikate zu sichern. Für die Kontextbindung im speziellen Anwendungsprofil würde es dann ausreichen, auf die entsprechende Ressource zu verweisen und ihr Zertifikat mitzusignieren, während eine zu signierende Dokumenteninstanz analog and das Profil zu binden ist. Enthält das Repository insbesondere Ressourcen und Anwendungsprofile, die von einer vertrauenswürdigen Institution evaluiert und zertifiziert wurden, ergibt sich aus ihrer Verwendung eine erhebliche Anhebung der Beweiseignung eines unter Verwendung eines Anwendungsprofils signierten Dokuments. Abbildung 1 zeigt eine mögliche Architektur.

Ein nicht ganz entferntes Beispiel für ein solches Repository mit einer Vielzahl an syntaktischen, funktionalen und Darstellungsbeschreibungen für digitale Dokumente ist das Comprehensive TeX Archive Network [CTAN]. Es stellt eine Bibliothek aller Grundbausteine dar, die man für Anwendungsprofile für TeX-Dokumente bräuchte. Seine verbreitete Verwendung in den Wissenschaften, aber auch durch kommerzielle Verleger, zeigt den prinzipiellen Nutzen (und möglichen Erfolg) solcher Repositories.

Anwendbarkeit von XML-Komponenten

Wir wollen an dieser Stelle kurz diskutieren, wie verschiedene existierende XML-Standards zur Eingrenzung der geschilderten Probleme eignen und wie sie sich gegebenenfalls in die skizzierte Infrastruktur für XML-Signatur Anwendungsprofile einordnen. Diese Darstellung erhebt aber keinen Anspruch auf Vollständigkeit und ist im wesentlichen als Anregung zu weiterer Forschung zu verstehen.

XML-Schema stellt nicht nur die Grundlage der Festlegung der Syntax und damit auch teilweise des zulässigen Inhalts signierbarer Dokumente in Anwendungsprofilen dar. Vielmehr können Schemata z.B. auch Einschränkungen an Signaturen nach XML-DSig darstellen (etwa: Signatur muss signierte Daten einhüllen, maximal N Gegenzeichnungen sind erlaubt, etc.), oder den Raum der möglichen Transformationen eines Dokuments beschneiden. Die Möglichkeit, Schemata mittels Inklusionsmechanismen hierarchisch aufzubauen, kann für Anwendungsprofile von entscheidendem Nutzen sein.

XSLT wurde bereits ausführlich bewertet. Ihr Hauptvorteil liegt letztlich in der Konformität zu einem offenen Standard und der prinzipiellen Menschenlesbarkeit und Bewertbarkeit von Transformationen und Darstellungen. Technische Vorteile, wie die verbreitete Unterstützung durch frei verfügbare Software legen ihre Verwendung als Kontextkomponenten nahe.

Stylesheet-Assoziation als generischer Mechanismus zur Bindung von Stylesheets an XML-Dokumente erreicht nicht die für ein XML-Signatur Anwendungsprofil nötige Kontextbindung, insbesondere ist die Integrität eines assoziierten Stylesheets nicht ohne weiteres zu sichern. In Anwendungsprofilen wird statt Stylesheet-Assoziation ein eigener Mechanismus zur Bindung des Darstellungs-Kontexts vorhanden sein.

XSL-FO [XSL-FO, jetzt XSL 1.0] hat **a**ls sehr detaillierte Seitenbeschreibungssprache große Vorteile bezüglich eindeutiger Darstellung, lässt jedoch einiges an Flexibilität missen, z.B. lassen sich hiermit keine Formularanwendungen darstellen. Jedoch kann XSL-FO sehr wohl einen Platz in Anwendungsprofilen haben, nämlich als statische Enddarstellung signierter Dokumente, z.B. zum Zwecke der späteren Begutachtung.

RDF. Obwohl RDF als generische Methode zur Zuordnung von Metadaten zu Dokumenten vordergründig keine Rolle in Bezug auf das Präsentationsproblem spielt, kann auch dieser Standard auf etwas subtilere Weise von Nutzen sein: Zumindest für die (nachgelagerte) Begutachtung signierter Dokumente sowie die (vorgelagerte) Evaluierung von Profilen ist es nötig, dem Bewerter außer den Kontextkomponenten auch zugehörige Hintergrunddokumente (Standards, Spezifikationen) zugänglich zu machen. Diesen Prozess zu automatisieren ist eine der genuinen Aufgaben von Metadaten.

Namespaces sind als Basis-Methode für jede komplexe XML-Anwendung und so auch für Signatur Anwendungsprofile unverzichtbar. [Schm_2000] gibt ein Beispiel für die Trennung von Zurechenbarkeiten, die auf der syntaktischen Ebene durch Namespaces zu erreichen ist.

5.Schlussfolgerungen

Fassen wir die Grundthesen unseres Diskussionsbeitrags zu XML-Signatur Anwendungsprofilen aus der Sicht des Präsentationsproblems noch einmal zusammen:

1. Digital signierte XML-Dokumente sind nicht *per se* beweiskräftiger als andere Datenformate, genauso wenig wie XML-Signaturen grundsätzlich geeigneter wären als andere Signaturformate. Die vielfältigen Möglichkeiten von XML, XML-DSig und zugehöriger Standards bergen vielmehr weitere Gefahren der fehlerhaften Anwendung und Präsentation. Diese Probleme sind nicht ohne weiteres durch eingeschränkte Verwendung des XML-Signatur Standards in den Griff zu bekommen.

2. XML-Standards, insbesondere Schemata und Stylesheets, können genutzt werden um Einschränkungen an die zu signierenden Daten, sowie an ihre Präsentation, in transparenter Weise zu beschreiben.

3. Solche Beschreibungen, also der **Kontext** eines signierten Dokuments müssen, um Sicherheit und Beweiskraft zu erzeugen, evaluiert und an das Dokument **gebunden** werden.

4. Die für die Kontextbindung erforderlichen Festlegungen sind nicht gut in einer Erweiterung des XML-DSig Standards unterzubringen, der dadurch überfrachtet würde, noch durch allgemeine Restriktionen von Signatur, Daten und Präsentation, die viele erwünschte Anwendungsfälle ausschließen würden, ohne weiteres in den Griff zu bekommen. Geeigneter erscheint die Spezifikation eines generischen Standards für „signierbare Dokumente“.

5. Auf Basis eines solchen Standards können anwendungsspezifische Spezifikationen geschaffen werden, die den Kern dessen bilden, was wir unter **XML-Signatur Anwendungsprofil** verstehen möchten.

6. Dieser Ansatz wäre zu komplementieren durch (öffentlich zugängliche) **Repositories**, die die notwendigen Ressourcen enthalten: Evaluierte Schemata, Stylesheets und Anwendungsprofile für verschiedenste Zwecke, sowie eventuell auch Metadaten und Dokumentationen.

7. Der potenzielle Nutzen von Repositories und Anwendungsprofilen ist es, zu ei-

ner Anhebung der Beweiseignung von signierten XML-Dokumenten beizutragen.

Literatur

[CTAN] Comprehensive TeX Archive Network, http://www.ctan.org

[ETSI_2002] ETSI (European Telecommunication Standards Institute): XML Advances Electronic Signatures (XAdES), ETSI TS 101 903 V1.1.1 (2002-2).

[Fox_1998] Fox, D.: Zu einem prinzipiellen Problem digitaler Signaturen, DuD 7/1998, 386 ff.

[Kunz _2001] Kunz, T.: Verbindliche Verträge in signiertem XML, Diplomarbeit, J. W. Goethe-Universität Frankfurt in Zusammenarbeit mit der GMD – Forschungszentrum Informationstechnik Darmstadt/Frankfurt 2001

[Pord_2003] Pordesch, U.: Die elektronische Form und das Präsentationsproblem, Baden-Baden 2003.

[RDF] W3C: Resource Description Framework (RDF) Model and Syntax Specification, W3C Recommendation 22 February 1999, http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/

[RDFS] W3C: Resource Description Framework (RDF) Schema Specification 1.0, W3C Candidate Recommendation 27 March 2000, http://www.w3.org/TR/2000/CR-rdf-schema-20000327/

[RFC 2630] Housley, R. (1999): Cryptographic Message Standard, June 1999

[Schm_2000] Schmidt, A. U.: Signiertes XML und das Präsentationsproblem, DuD 3/2000, 153 ff.

[Stylesheet_Ass] W3C: Associating Style Sheet with XML documents Version 1.0, W3C Recommendation 29 June 1999, http://www.w3.org/TR/xml-stylesheet/

[XML-DSig] W3C: XML-Signature Syntax and Processing, W3C Recommendation 12 February 2002, http://www.w3.org/TR/xmldsig-core/.

[XSL-FO] W3C: Extensible Stylesheet Language (XSL) Version 1.0, W3C Recommendation 15 October 2001, http://www.w3.org/TR/xsl

[XPath] W3C: XML Path Language (XPath) Version 1.0, W3C Recommendation 16 November 1999/TR/xpath/

[XPath-Filter] XML-Signature XPath Filter 2.0, W3C Recommendation 08 November 2002, http://www.w3.org/TR/xmldsig-filter2

[XML-Schema_1] W3C: XML Schema Part 1: Structures, W3C Recommendation 2 May 2001, http://www.w3.org/TR/xmlschema-1/

[XML-Schema_2] W3C: XML Schema Part 2: Datatypes, W3C Recommendation 2 May 2001, http://www.w3.org/TR/xmlschema-2/

[XSL-T] W3C: XSL Transformations (XSLT) Version 1.0, W3C Recommendation 16 November 1999, http://www.w3.org/TR/xslt