eCH-0091: Best Practices zu XML-Signatur und Verschlüsselung

Name	Best Practices zu XML-Signatur und Verschlüsselung
Standard-Nummer	eCH-0091
Kategorie	Best Practice
Reifegrad	Definiert
Version	1.00
Status	Aufgehoben
Genehmigt am	2009-11-26
Ausgabedatum	2009-10-15
Ersetzt Standard	
Sprachen	Deutsch
Autoren	Fachgruppe XML Daniel Muster (Initiant dieses Themas) Willy Müller, Informatikstrategieorgan Bund (ISB) Claude Eisenhut, Eisenhut Informatik Alexander Pina, Unisys (Schweiz) AG Eric Dubuis, Berner Fachhochschule Gilles Maitre Stephan Fischli, Berner Fachhochschule
Herausgeber / Vertrieb	Verein eCH, Mainaustrasse 30, Postfach, 8034 Zürich T 044 388 74 64, F 044 388 71 80 www.ech.ch / info@ech.ch

Zusammenfassung

XML-Objekte können mittels für XML speziell standardisierten Methoden (Signaturen, Verschlüsselungen) geschützt werden. Probleme treten dann auf, wenn ganze Dokumente inklusive z.B. darin verwiesener Bilder, Schemas oder Informationen zur Darstellung (Layout) auch geschützt werden müssen.

Das hier vorliegende Dokument will einerseits auf die damit verbundenen Probleme hinweisen, Lösungsvorschläge dazu unterbreiten und andererseits die bestehenden Standards zu XML auf die besonderen Gegebenheiten des Schweizerischen eGovernment anpassen. Der Fokus liegt dabei auf dem Schutz von Verwaltungsdokumenten auf Basis von XML.

Inhaltsverzeichnis

1 Status des Dokuments

Aufgehoben: Das Dokument wurde von eCH zurückgezogen. Er darf nicht mehr genutzt werden.

2 Einleitung

2.1 Ziel des Dokuments

Dieses Dokument will einerseits auf (Sicherheits)probleme hinweisen, welche bei der Verschlüsselung und bei der Signierung von XML-Dokumenten auftreten können, und etwelche Lösungsvorschläge zur Behebung der besagten Probleme unterbreiten; dies aber mit dem Fokus auf XML-Verwaltungsdokumente.

Verwaltungsdokumente zeichnen sich dadurch aus, dass sie losgelöst von jeglicher Datenkommunikation oder jeglichem Datenaustausch betrachtet werden können und sollten; dies zum Beispiel im Unterschied zu einer SOAP Message, einer von vielen XML-Anwendungen.

Bei der Signatur und Verschlüsselung von XML-Verwaltungsdokumenten geht es darum, dass sie vollständig (d.h. mit allen sicherheitsrelevante Informationen enthalten) signiert und verschlüsselt werden, dann entschlüsselt und wieder zusammengesetzt werden können, so dass die Signatur ohne Änderungen und innerhalb einer vorgesehenen Zeit nicht an Gültigkeit verliert.

In den folgenden Kapiteln liegt der Hauptfokus auf XML-Verwaltungsdokumenten:

- Kapitel 3 „Erläuterung der Problematik“
- Kapitel 4 „Risiken und Massnahmen“
- Kapitel 5 „Präzisierung der bestehenden Standards“
- Kapitel 6 „Alternativlösung“
- Anhang A
- Anhang B

2.2 Terminologie der Empfehlungen

Richtlinien in diesem Dokument werden gemäss der Terminologie aus [RFC 2119] angegeben, dabei kommen die folgenden Ausdrücke zur Anwendung, die durch **GROSSSCHREIBUNG** als Wörter mit den folgenden Bedeutungen kenntlich gemacht werden (Zitat aus [RFC 2119]):

- **MUST**: This word, or the terms "**REQUIRED**" or "**SHALL**", mean that the definition is an absolute requirement of the specification.
- **MUST NOT**: This phrase, or the phrase "**SHALL NOT**" mean that that definition is an absolute prohibition of the specification.

- **SHOULD**: This word, or the adjective "**RECOMMENDED**", mean that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.
- **SHOULD NOT**: This phrase, or the phrase "**NOT RECOMMENDED**" mean that there may exist valid reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.
- **MAY**: This word, or the adjective "**OPTIONAL**", mean that an item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because the vendor feels that it enhances the product while another vendor may omit the same item. An implementation which does not include a particular option MUST be prepared to interoperate with another implementation which does include the option, though perhaps with reduced functionality. In the same vein an implementation which does include a particular option MUST be prepared to interoperate with another implementation which does not include the option (except, of course, for the feature the option provides.)

3 Erläuterung der Problematik

Im Standard [CWA 14170], resp. [CWA 14171] wird empfohlen, welche Sicherheitsmassnahmen beim Erstellen, resp. beim Verifizieren einer elektronischen Signatur getroffen werden sollen. Das hier vorliegende Dokument im Gegensatz zu den erwähnten CWA Standards beschränkt sich hauptsächlich auf den folgenden Aspekt:

> „What you see is what you will sign or what you verify is what you see." Dies auch lediglich im Kontext von XML-Objekten. Was darunter genauer zu verstehen ist, wird rudimentär in den nächsten Unterkapiteln erläutert.

Im Unterschied zu den erwähnten Standards werden hier auch noch die Probleme der Verschlüsselung von XML-Dokumenten erläutert, insbesondere die Verschlüsselung von signierten XML-Dokumenten und -Objekten.

3.1 Probleme bei der XML-Signatur

Bei der XML-Signatur besteht das Problem, dass gegebenenfalls nicht das ganze Dokument, inklusive allfälliger Unterobjekte wie Schemas, CSS File, oder der darin möglicherweise enthaltenen Bilder signiert wird, sondern lediglich das Hauptobjekt. Somit besteht die Möglichkeit, dass das Dokument in seiner Erscheinung (Präsentation) verändert werden kann, ohne dass die Signatur unter dem Hauptobjekt an Gültigkeit verliert, z.B. indem etwelche Unterobjekte verändert oder ersetzt werden. Diese Tatsache birgt erhebliche Sicherheitslücken.

Mögliches Angriffsszenario: Carl fertigt ein Dokument mit dem Hauptobjekt A z.B. im XML- oder HTML-Format an. Wie sich die Schriften (der Inhalt des Objekts) am Bildschirm präsentieren sollen, definiert er in einem Unterobjekt L, z.B. in einer CSS Datei. Das Hauptobjekt A und das Unterobjekt L sind über einen internen Verweis im Hauptobjekt A verbunden.

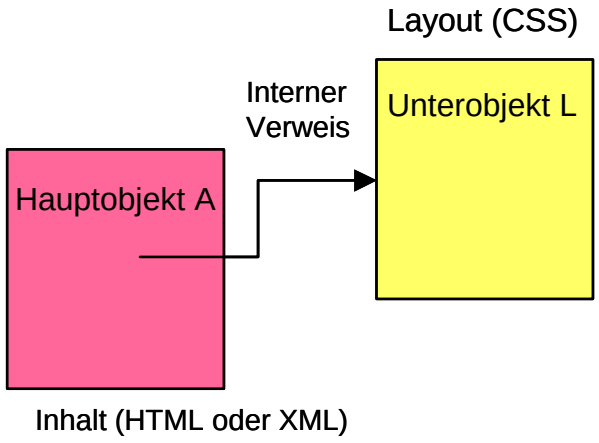


Abb. 3-1 Trennung von Inhalt und Darstellung bei einem Dokument

Carl hat die Layoutinformation im Unterobjekt L (z.B. im CSS Format) so definiert, dass gewisse Passagen mit weisser Schrift auf weisem Hintergrund präsentiert werden. Er unterbreitet das Dokument mit der Layoutinformation L Alice. Alice signiert aber lediglich das Hauptobjekt A, jedoch nicht auch noch das Unterobjekt L (z.B. das CSS File) und sendet das signierte Hauptobjekt A an Carl zurück.

Carl verwandelt das Unterobjekt L in L', so dass die weissen Textpassagen und Worte sich nun mit schwarzer Schrift auf weissem Grund am Bildschirm präsentieren. Somit kann er ein für ihn vorteilhaftes, von Alice signiertes „Dokument" präsentieren, welches in dieser Form von Alice nicht gesehen worden ist. Dies ist möglich, weil nur das Hauptobjekt und nicht auch noch die dazu gehörigen Unterobjekte signiert worden sind.

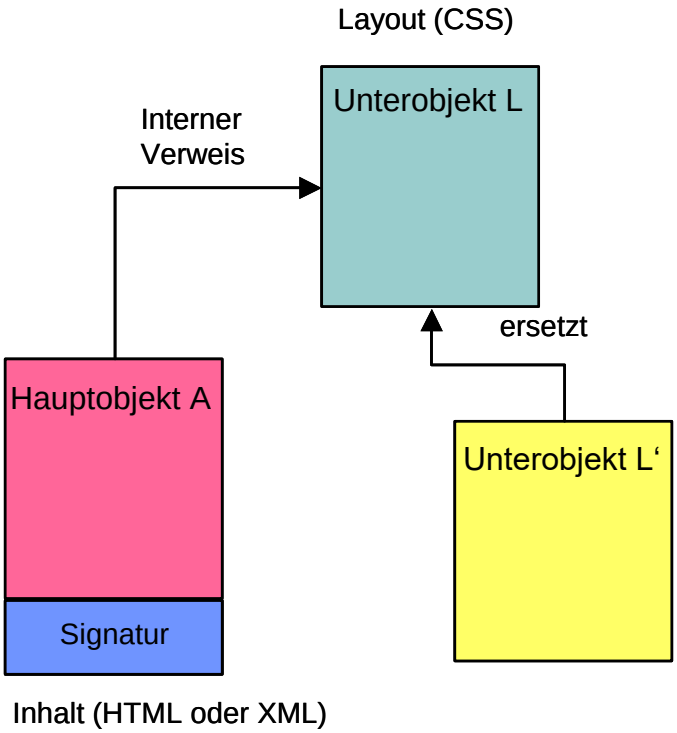


Abb. 3-2 Auswechslung von Inhalten, wobei die Gültigkeit der Signatur erhalten bleibt, doch das Dokument sich anders präsentiert.

Bei signierten Dokumenten darf sich die Präsentation oder das Erscheinungsbild nicht verändern, ohne dass die Signatur dabei ungültig wird.

Wann ein Dokument signiert werden soll, hängt von den jeweiligen Sicherheitsanforderungen an die Authentizität, Integrität, Verbindlichkeit und Nachvollziehbarkeit ab. Dieses Dokument macht aber keine Aussage dazu, welche Sicherheitsanforderungen gelten und folglich wann ein Dokument zu signieren ist.

Es gibt z.B. weitere Szenarios oder Prozesse welche eine Signatur des vollständigen Dokuments und nicht nur des Hauptobjekts und eine vollständige Präsentation des signierten Dokuments erfordern.

Das zuvor skizzierte Szenario des Missbrauchs der elektronischen Signatur basiert auf Absicht und wurde zwecks besserer Illustration und Verständnis für das hier zu behandelnde Problem präsentiert. Dass gewisse Passagen des Dokuments ausgewechselt, verändert oder gelöscht werden und folglich nicht mehr wie vorgesehen richtig rekonstruierbar sind und geprüft werden können, kann auch aus Unachtsamkeit und Fahrlässigkeit geschehen.

3.2 Probleme bei der Verschlüsselung

Probleme treten bei der XML-Verschlüsselung gemäss W3 Standard eines (Verwaltungs)Dokuments in XML auf: Sei denn, dass Teile (Unterobjekte) des Dokuments nicht vorliegen, oder Unterobjekte des Dokuments wie z.B. Bilder unbeabsichtigt als Klartext übertragen werden. Das ganze Dokument mit all seinen Unterobjekten sollte verschlüsselt werden, damit verhindert wird, dass einerseits sensitive Inhalte in den Unterobjekten im Klartext vorliegen und eingesehen werden können, andererseits aus den unverschlüsselten Unterobjekten Rückschlüsse auf die sensitiven Inhalte der verschlüsselten Objekte gezogen werden können.

3.3 Modell

Im Unterschied zu den beiden Standards [CWA 14170] und [CWA 14171] wird hier ein viel einfacheres Modell zur Bildung und Verifikation der Signatur vorgestellt, weil hier nur ein kleinerer Aspekt behandelt werden soll.

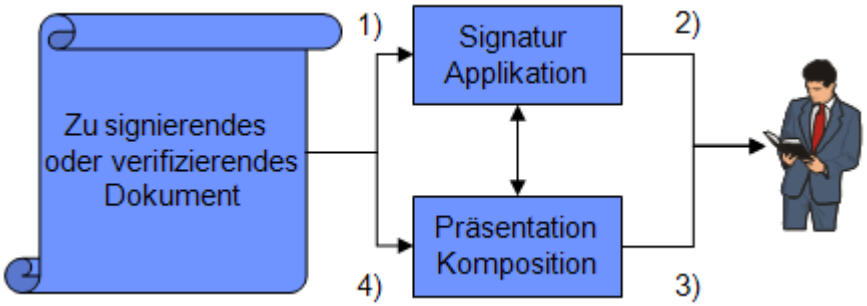


Abb. 3-3 Modell für die Erstellung und Verifikaton der Sigantur

Die Signatur Applikation verifiziert und signiert (1) die Objekte und stellt das Ergebnis dem Benutzer zur Verfügung (2). Die Rekonstruktion oder Komposition setzt das Dokument zusammen, welches vom Benutzer (3) zu signieren oder zu verifizieren ist (4). Anmerkung: Die Rekonstruktion und Anzeige des Dokuments können aber auch Bestandteil der Signatur Applikation sein. Wenn nicht, sollte die Komposition von der Signatur Applikation gestartet und dadurch das entsprechende zu signierende Dokument angezeigt werden können.

Das Dokument selber kann sich neben dem Hauptobjekt aus mehreren weiteren Unterobjekten zusammensetzen, wie:

- Bilder
- Schema
- (Indirekt) referenzierter Code
- Anweisungen zu einer Transformation des Dokuments
- Formatierungsanweisungen
- Andere XML-Objekte oder Bestandteile davon

Was die Komposition leisten muss, wie und was präsentiert werden soll, hängt hauptsächlich davon ab, wie die Kommunikationsgesellschaft dies zuvor definierte. Damit aber wirklich das ganze Dokument mit der Signatur des Benutzers geschützt wird, **muss** mindestens alles signiert werden, was für die korrekte Komposition und Präsentation des Dokuments relevant ist.

Wie nun ein Dokument rekonstruiert und vor allem wie das Dokument korrekt oder geschützt präsentiert wird, liegt ausserhalb der Zielsetzung dieses eCH-Dokuments. Dieses eCH-Dokument will lediglich Ergänzungen anfügen und Empfehlungen darüber abgeben, wie das gesamte XML-Dokument mit einer XML-Signatur signiert und vollständig bezüglich der Vertraulichkeit geschützt werden kann.

4 Risiken und Massnahmen

Die Anwendungsfälle zu den hier nun untersuchten Risiken werden unterteilt in

- XML-Signatur
- XML-Verschlüsselung
- XML-Signatur mit Verschlüsselung

In den genannten Fällen wird unterschieden, ob ein Dokument voll automatisiert verarbeitet oder das Anbringen der elektronischen Signatur oder Verschlüsselung durch einen Benutzer veranlasst wird.

Der XML-Signatur Standard [RFC 3275] erlaubt die folgenden 3 Arten von XML-Signaturen:

- **Detached**: Die Signatur zeigt entweder auf ein XML-Element ausserhalb der XML-Hierarchie, in welche das XML-Signaturelement eingebettet ist, oder auf eine beliebige externe Datei, welche mittels URI referenziert werden kann. (Bei diesem Verfahren zeigt die Referenz der Signatur auf ein XML-Element, welches sich nicht entlang des Pfades vom Signatur-Element zur Wurzel des Dokuments befindet).

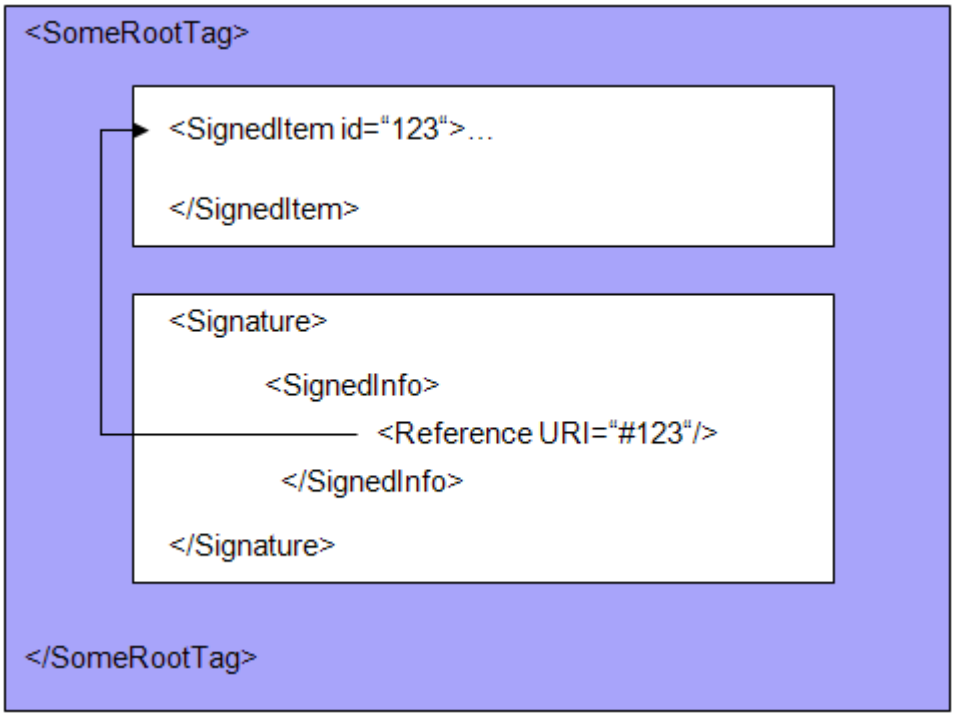


Abb. 4-1 Detached Signature (erste Ausprägung)

Bei diesem Verfahren ist es auch möglich, dass die Referenz der Signatur auf eine Ressource zeigt, welche sich ausserhalb des XML-Dokuments befindet.

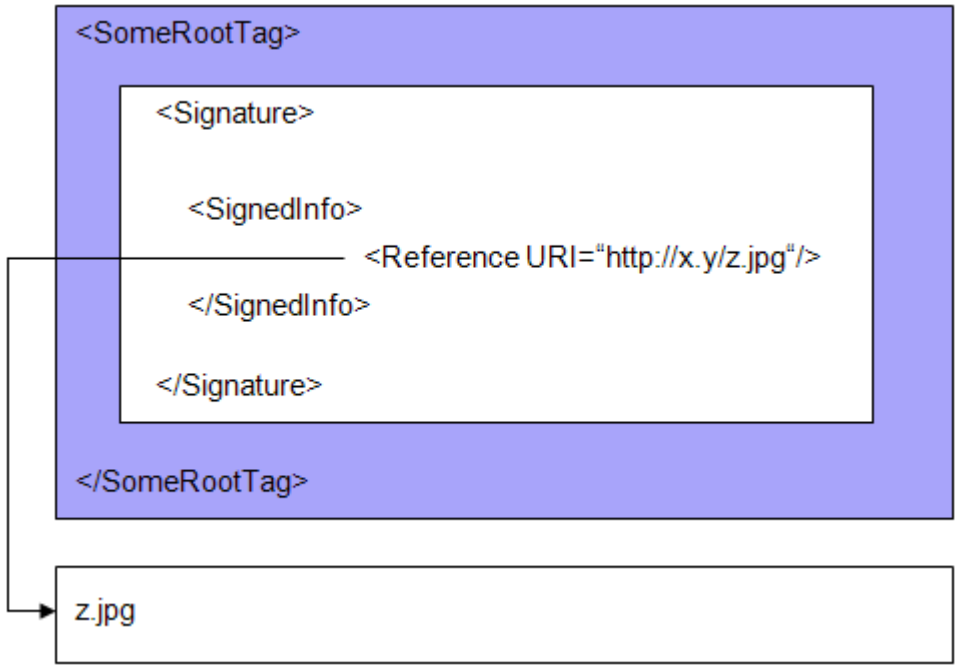


Abb. 4-2 Detached Signature (zweite Ausprägung)

- **Enveloped**: Die Signatur zeigt auf ein Elternelement in der XML-Hierarchie, in welche das XML-Signaturelement eingebettet ist. (Bei diesem Verfahren zeigt die Referenz der Signatur auf ein Eltern-XML-Element der Signatur).

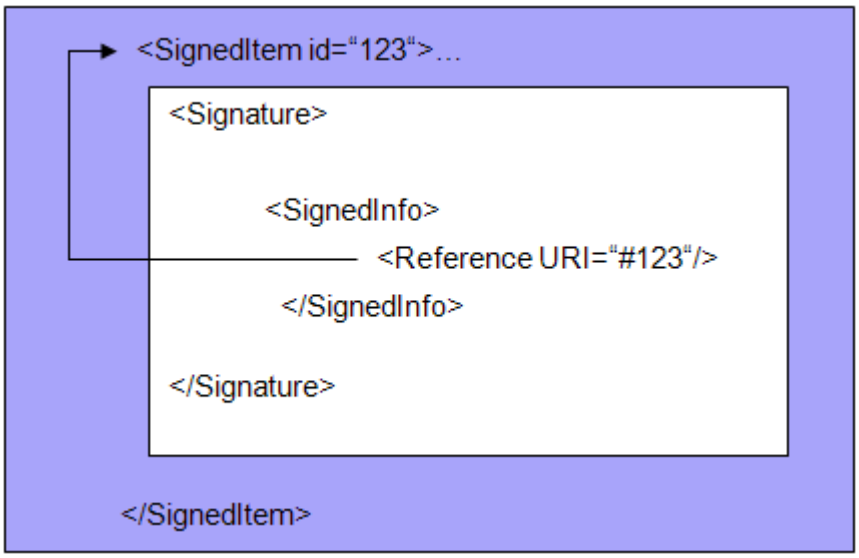


Abb. 4-3 Enveloped Signature

- **Enveloping**: Die Signatur enthält die Information, welche signiert wurde, als Kindelement des XML-Signaturelements (Bei diesem Verfahren wird die Information, die zu signieren ist, in die Signatur eingepackt).

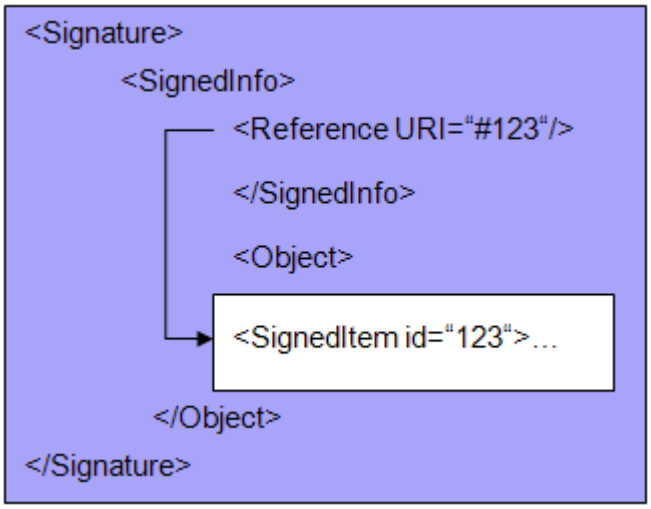


Abb. 4-4 Enveloping Signature

Eine XML-Signatur kann eine Kombination von detached, enveloped und auch enveloping sein.

Der W3C Standard zur XML-Verschlüsselung erlaubt die folgenden 3 Arten von XML-Verschlüsselungen:

- Die Verschlüsselung des Inhalts eines XML-Elements
- Die Verschlüsselung des XML-Elements und dessen Inhalt
- Die Verschlüsselung von irgendwelchen Objekten, welche auch XML-Bestandteile haben können.

4.1 Risiken

Es gibt eine Fülle von weiteren als hier aufgelisteten Risiken im Kontext der Verschlüsselung und Signatur. Doch die Massnahmen hierzu liegen ausserhalb der Zielsetzung dieses Dokuments.

4.1.1 Signatur

Bei der Signatur kann das Risiko bestehen, dass nicht alle sicherheitsrelevanten Objekte des Dokuments signiert werden. Z.B. werden nur das Hauptobjekt, aber nicht auch die intern verwiesenen Objekte signiert. Somit können schützenswerte Teile (Unterobjekte) des Dokuments und dessen Erscheinungsbild verändert werden, ohne dass dabei die Signatur unter dem vermeintlichen Dokument ungültig wird.

4.1.2 Verschlüsselung

Bei der Verschlüsselung besteht das Risiko, dass nicht alle Objekte des Dokuments verschlüsselt werden. Z.B. werden nur das Hauptobjekt, aber nicht auch die intern verwiesenen Unterobjekte verschlüsselt. Somit können sensitive Informationen in den Unterobjekten ungewollt offen gelegt werden und gegebenenfalls aus den unverschlüsselten Unterobjekten Rückschlüsse auf das verschlüsselte Objekt gemacht werden.

4.2 Signatur

Anmerkung zur folgenden Tabelle: Unter Anwendungsfälle werden Möglichkeiten aufgeführt, wie die im Kapitel 4.1.1 erwähnten Risiken auftreten können.

4.2.1 Signatur vom Benutzer ausgelöst

Nr.	Anwendungsfälle	Massnahme	Bemerkung
1	**Versand von signierten Dokumenten**. Nicht das ganze Dokument wird signiert, sondern nur Teile davon oder nur das Hauptobjekt. Konsequenz ist, dass wesentliche Bestandteile des Dokuments wie die intern verwiesenen Unterobjekte ausgewechselt werden können, ohne dass dabei die Signatur an Gültigkeit verliert.	**MUST:** Mindestens alle intern verwiesenen Objekte müssen Bestandteil der Signatur (von der Signatur erfasst) sein.	Zur Behandlung von Verweisen, s. auch Kapitel 5.1.3
2	**Ausfüllen eines Formulars über eine online Verbindung.** Ein Formular wird online am Bildschirm ausgefüllt. Die dabei gemachten Angaben werden signiert und dann online dem Server zur weiteren Verarbeitung übermittelt. Hier besteht einerseits das Problem, dass der Benutzer nicht vollständig erkennen kann, was er unterschreibt, und unter Umständen besteht das Problem, dass er nicht archivieren kann, was er unterschrieben hat.	Folgende Alternativen stehen zur Verfügung: 1. **MUST:** Alle intern verwiesenen Objekte müssen Bestandteil der Signatur (von der Signatur erfasst) sein. 2. **MUST:** Man muss das XML-Dokument vollständig in ein *PDF/A* Objekt umwandeln. Dieses *PDF/A* Objekt muss dann vom Benutzer nach PKCS#7 signiert werden.	Zur Behandlung von Verweisen, s. auch Kapitel 5.1.3

Nr.	Anwendungsfälle	Massnahme	Bemerkung
3	**Versand von signierten Dokumenten mit Programmcode**. In XML, aber vor allem in Dokumenten mit einem Hauptobjekt in HTML, kann ausführbarer Code wie ActiveX, JavaScript oder Java Applets enthalten sein. Hierbei können die Parameter geändert und dabei das Erscheinungsbild des Dokuments verändert werden, ohne dass dabei die Gültigkeit der Signatur ändert.	**MUST:** Vor der Signatur muss die Sicherheitsapplikation eine Warnung herauszugeben, dass das betreffende Verwaltungsdokument Programmcode enthält. **SHOULD NOT:** Ein Dokument mit solchen Inhalten sollte vom Benutzer nicht signiert werden.	**Anmerkung:** Der hier beschriebene Fall sollte von der Anwendung „Code Signing“ unterschieden werden. Beim Code Signing will man die Herkunft des Programms belegen, indem man das Programm selber signiert. Hier will man aber verhindern, dass ungewollt Änderungen am (Verwaltungs)Dokument vorgenommen werden können, wobei die Signatur unter dem Dokument auch nach den Änderungen weiterhin gültig bleibt. S. auch Kapitel 5.1.2 „Enthaltene Programme im Dokument“.

4.2.2 Voll automatisierter Prozess

	Anwendungsfälle	Massnahme	Bemerkung
1	**Authentizität von Dokumenten**	**MUST:** Die für die Authentizität und Integrität relevante Information muss signiert werden, d.h. alle internen Verweise auch.	Bei den voll automatisierten Prozessen können andere Verfahren als eine Signatur für den Schutz der Authentizität der XML-Dokumente eingesetzt werden, als bei Prozessen, welche vom Benutzer initiiert werden oder für einen Benutzer bestimmt sind, siehe dazu Kapitel 5.2.6.3 „HMAC“.

4.3 XML Verschlüsselung

Anmerkung zur folgenden Tabelle: Unter Anwendungsfälle werden Möglichkeiten aufgeführt, wie die im Kapitel 4.1.2 erwähnten Risiken auftreten können.

4.3.1 Verschlüsselung wird vom Benutzer ausgelöst

Nr.	Anwendungsfälle	Massnahme	Bemerkung
1	**Verschlüsseln von XML-Dokumenten**. Nicht das ganze Dokument wird verschlüsselt, sondern nur Teile davon oder nur das Hauptobjekt. Konsequenz ist, dass unter Umständen Teile des Dokuments im Klartext übertragen werden. Dies würde Rückschlüsse auf die verschlüsselten Objekte erlauben.	**MUST:** Alle intern verwiesenen sensitiven Objekte müssen zusammen mit dem Hauptobjekt verschlüsselt werden.	

Nr.	Anwendungsfälle	Massnahme	Bemerkung
2	**Ausfüllen eines Formulars über eine verschlüsselte online Verbindung.**	**MUST:** Es muss beachtet werden, dass alle Objekte des Formulars und alle gemachten Angaben verschlüsselt werden.	Es besteht das Problem, dass nicht alle Objekte des Dokuments über die SSL/TLS Verbindung verschlüsselt übertragen, sondern über einen separate unverschlüsselte HTTP Verbindung ausgetauscht werden

4.3.2 Voll Automatisierte Verschlüsselung von XML-Dokumenten

Nr.	Anwendungsfälle	siehe	Bemerkung
1	**Austausch von verschlüsselten XML-Dokumenten**.	Zwei gleichwertige Massnahmen stehen zur Verfügung: 1. **MUST:** Alle zum Dokument gehörenden Objekte, worauf intern verwiesen wird, müssen verschlüsselt werden. 2. **MUST:** Der Austausch, z.B. die Kommunikationsverbindung muss verschlüsselt werden. Alle Objekte, worauf intern verwiesen wird, müssen dabei auch verschlüsselt werden.	Im automatisierten Datenaustausch ist es nicht erforderlich, dass alle Informationen zum Dokument wie z.B. die Schemas jedes Mal mitgereicht werden.

4.4 Signatur mit Verschlüsselung

Es wird angenommen, dass das Dokument verschlüsselt wird und nicht die Kommunikationsverbindung. Zudem sollte geklärt werden, ob das Dokument zuerst signiert und dann chiffriert werden soll oder umgekehrt.

Nr.	Anwendungsfälle	Massnahme	Bemerkung
1	**Austausch von verschlüsselten und signierten XML-Dokumenten**. Wird zuerst verschlüsselt und dann signiert, so glaubt man zu wissen, woher die Meldung stammt und dass keine Änderungen (auf dem Transportweg) vorgenommen worden sind. Man würde eine Änderung sofort feststellen, weil der Wert der Hashfunktion unterschiedlich ist. Wird aber zuerst signiert und dann verschlüsselt, kann man z.B. belegen kann, was signiert worden ist. Zudem kann die Meldung unverschlüsselt, aber signiert bei beiden Parteien abgelegt und aufbewahrt werden. Beim Empfang der Dokumente kann aber keine Vorselektion (z.B. auf Spam) vor der Entschlüsselung getroffen werden, weil man nicht weiss, von wem das Dokument stammt. Z.B. kann Unsinn (z.B. Werbung) für den Empfänger verschlüsselt worden sein. (Dass eine Meldung verschlüsselt worden ist, bedeutet noch lange nicht, dass diese Meldung vertrauliche Informationen enthält.)	**MUST:** Für (rechtlich) verbindliche Dokumente muss wegen der Archivierung die verbindliche Signatur vor der Verschlüsselung angefertigt werden. **SHOULD:** Das Dokument sollte zuerst signiert, dann verschlüsselt und nachträglich wieder signiert werden. **SHOULD NOT:** Wenn ein verschlüsseltes Objekt signiert wird, dann sollte die Signatur nicht qualifiziert sein. **Grund:** Eine qualifizierte Signatur hat eine rechtliche Implikation. Folglich besteht im Allgemeinen das Bedürfnis aufzubewahren und später lesen und darlegen zu können, was man verbindlich signiert hat. Ist die Datei verschlüsselt, besteht diese Möglichkeit unter Umständen nicht mehr.	

Nr.	Anwendungsfälle	Massnahme	Bemerkung
2	**Austausch von verschlüsselten und signierten XML-Dokumenten**. Der Einsatz von Detached Signature, welche nicht verschlüsselt werden, erlaubt u.a. eine beschleunigte Brute Force Attacke, weil eine beschleunigte Plausibilitäts-prüfung für einen Kandidaten eines Entschlüsselungsschlüs-sels durchgeführt werden kann, siehe dazu [Mud] S. 204.	**SHOULD:** Man sollte auch die Detached Signature verschlüsseln, wenn das dazu gehörige Dokument oder Objekt chiffriert wird.	

5 Präzisierung der bestehenden Standards

In diesem Kapitel werden weitere Angaben zu den verschiedenen Massnahmen der XML-Signatur und der XML-Verschlüsselung gemacht. Dabei wird auf die entsprechenden Standards der IETF und W3C abgestützt. Grundsätzlich sind diese Standards als verbindlich zu erachten. Jedoch werden hier gewisse technische Aspekte zu den Standards ergänzt und falls nötig auf die Schweizerischen Gegebenheiten angepasst, d.h. gegebenenfalls eine zu den Standards unterschiedliche Empfehlung abgegeben.

Es werden hier nur dazu Angaben gemacht, wo die FG eine zu den Standards unterschiedliche Meinung vertritt oder wo der Standard etwas nicht genau oder nicht definiert hat.

Wichtig: Bei der IT-Sicherheit von Applikationsdaten lässt sich die Sicherheit nicht von der Applikation trennen, s. auch Kapitel 3 und 4 [SOAP Security with Attachements]. Dies geht auch aus den folgenden Ausführungen hervor.

Die Reihenfolge der folgenden Unterkapitel orientiert sich an dem wie folgt beschriebenen Ablauf, wie XML-Objekte und Dateien geschützt werden:

- Das Dokument wird gegebenenfalls vorverarbeitet, bevor etwelche Signaturen auf das Dokument angewandt werden. Warum dies notwendig sein kann, geht aus dem Kapitel 5.1 hervor.
- Das Dokument wird signiert, falls erforderlich.
- Das Dokument wird dann gegebenenfalls verschlüsselt.
- Das verschlüsselte Dokument wird gegebenenfalls signiert.
- Das Dokument wird dem Empfänger zugestellt.
- Die Signatur um das Dokument wird geprüft.
- Das Dokument wird entschlüsselt.
- Die Signatur unter dem Dokument wird geprüft.
- Das Dokument wird separat wieder zusammengestellt.

Wie die einzelnen, oben aufgeführten Schritte genau ablaufen, ist im entsprechenden RFC und W3C Standard beschrieben.

5.1 Vorverarbeitung des Dokuments

5.1.1 Separierung des Dokuments

Verschachtelung von Signaturen innerhalb desselben Dokuments will man möglichst verhindern.

SHOULD: Die zu signierenden Objekte sollten, falls möglich, separiert werden, d.h. in Teilobjekte zerlegt werden, damit nicht Verschachtelungen von Signaturen entstehen. Dabei sollte auch beachtet werden, dass das Dokument bereits in einer nach XML genormten Form vorliegt.

Grund: Verschachtelte Signaturen erschweren die Prüfung und die Archivierung elektronisch signierter Dokumente.

5.1.2 Enthaltene Programme im Dokument

Hier will man verhindern, dass Änderungen am Dokument vorgenommen werden können, ohne dass die Signatur unter dem Dokument auch nach einer Änderung weiterhin gültig bleibt. Eine Signatur schützt sowohl Herkunft als auch Integrität des Dokumentes.

SHOULD: Zu signierende (Verwaltungs)dokumente sollten so gestalten sein, dass sie im betreffenden Kontext oder Anwendung keine Programme (wie Makros, usw.) enthalten.

Im entsprechenden Kontext sind die Dokumente auf etwelche Programminhalte in XML-Dokumenten zu prüfen. Die Prüfung sollte mit Programmen verifiziert werden, welche für die entsprechende Anwendung zertifiziert sind. Damit aber eine solche Prüfung sinnvoll vorgenommen werden kann, ist für die Anwendung eine Beschreibung mit entsprechendem Schema herzustellen.

Falls solche Programme enthalten sind, sollte bei nicht voll automatisierten Prozessen eine entsprechende Warnmeldung dem Benutzer angezeigt werden.

SHOULD NOT: Sind Programme im Dokument enthalten, sollte keine Signatur erstellt und der Prozess der Signaturherstellung abgebrochen werden.

Grund: Die Programme innerhalb des Dokuments können das Dokument so verändern, dass es sich nach der Signatur anders präsentiert, als bei der Verifikation der Signatur, wobei die Signatur aber weiterhin gültig bleibt.

5.1.3 Behandlung der internen Verweise

Es soll erreicht werden, dass das signierte Dokument an verschiedenen Orten und zu verschiedenen Zeiten wieder aus seinen Bestandteilen (Objekten) zusammengesetzt werden kann.

MUST: Bevor mit dem Prozess der Signatur begonnen wird, muss darauf geachtet werden, dass sämtliche im Dokument vorhandenen internen Verweise relativ sind und nicht absolut aufgeführt werden.

Grund: Nur so ist es dem Empfänger möglich, das Dokument, wie es signiert worden ist, zusammenzustellen, ohne dabei etwas zu verändern. Wäre z.B. ein interner Verweis absolut aufgeführt und zeigt dieser auf ein Verzeichnis beim Versender, so hat dies folgende Nachteile:

- Der Empfänger kann das Dokument nicht wie signiert wieder zusammenstellen und präsentieren lassen, denn unter Umständen hat er keine Berechtigung auf den Ort, wo das verwiesene Objekt gespeichert worden ist.
- Falls er doch Zugriff auf die verwiesenen Objekte haben sollte, wird das verwiesene Objekt dann möglicherweise im Klartext übermittelt, wenn das Dokument dem Empfänger präsentiert wird.
- Der Empfänger kann das Dokument nicht wie signiert vollständig speichern und aus den so gespeicherten Daten später wieder zusammenstellen, ohne dass er etwas am Dokument ändert. Etwelche Änderungen am Dokument haben aber eine ungültige Signatur zur Folge.

5.2 Signaturerstellung

Der Prozess der Signaturerstellung ist im Kapitel 3 des Standards [RFC 3275] beschrieben. Eine bildliche Darstellung dieses Prozesses ist in Anhang A dargestellt.

5.2.1 Signaturtypwahl

Bekanntlich stehen 3 Typen von XML-Signaturen zur Verfügung. Es gibt auch Mischformen dazu.

MAY: Der Typ Enveloping oder Detached ist bei der Signatur über mehr als ein Objekt zu verwenden.

Grund: Detached Signature können im Falle einer Archivierung separat und losgelöst vom Dokument behandelt werden.

5.2.2 Signatur

Wenn nur Teile des Dokuments signiert werden, dann können die anderen Teile ersetzt werden, ohne dass die Signatur ihre Gültigkeit verliert, und folglich wird der Integritätsschutz dabei beeinträchtigt.

MUST: Die Signatur muss sich über alle sicherheitsrelevanten Teile des Verwaltungsdokuments erstrecken.

5.2.3 Transformation der Objekte

Die Transformation wird im Standard [RFC 3275] als die Bearbeitung des Objekts bezeichnet, bevor der Hashwert (Message Digest oder die kryptographische Prüfsumme) über das transformierte Objekt generiert wird, s. auch Anhang A (vereinfachte bildliche Darstellung der XML-Signatur).

MUST: Eine Kanonisierung der XML-Unterobjekte und des XML-Hauptobjekts muss durchgeführt werden. Für externe binäre Daten (engl. Binaries) wie Bilder ist eine Base64 Codierung anzuwenden.

SHOULD: Die Kanonisierung, welche auf die XML-Objekt angewandt wird, *sollte* mit der Kanonisierung für die Liste der Elemente der Prüfsummen identisch sein.

Grund: Vereinfacht ausgedrückt, die kryptographische Prüfsummenbildung zur Bildung der Signatur ist so stark, wie der schwächste eingesetzte Algorithmus zur Bildung der Message Digest.

Im Unterschied zum Standard [RFC 3275] werden noch die folgenden zwei Kanonisierungen empfohlen:

- Exklusive Kanonisierung mit Kommentar, s. [RFC 3741]
- Exklusive Kanonisierung ohne Kommentar, s. [RFC 3741]

MUST NOT: Sowohl XSLT als auch weitere im Standard aufgeführte Verfahren dürfen aus Sicherheitsüberlegungen nicht angewandt werden.

Anmerkung zu den Transformationen: Bei den Transformationen gilt es zu unterscheiden:

1. Definierte Transformationen wie Base64 Codierung, die in den Standards [RFC 3741] und [RFC 3076] erwähnten Kanonisierungen.
2. Transformationen, deren Verhalten im Dokument konfigurierbar ist
3. Transformationen, deren Verhalten wohl konfigurierbar ist, aber die Konfiguration ausserhalb des Dokumentenkontexts definiert wird.

Die letzten 2 Transformationstypen (2,3) bilden für die Signatur ein Sicherheitsproblem, weil das Ergebnis der Transformation schwer kontrollierbar ist und somit je nach Ergebnis jede Signaturprüfung erfolgreich verläuft. Beispiel: Man transformiert jedes Objekt auf einen bestimmten Text T. Folglich wird für diesen Text T eine Prüfsumme angefertigt. Ändert man das XML-Objekt, dann bleibt die Signatur weiterhin gültig, weil auch dieses Objekt auf den zuvor definierten Text transformiert wird und sich somit die daraus resultierende Prüfsumme nicht ändert.

Weiterer Grund: S. Kapitel 8.1 im Standard [RFC 3275]

Im Dokument [SOAP Security] von OASIS, S. 36 Rz 1185 ff., sind Pro und Contra inklusive und exklusive Kanonisierung aufgeführt. Vereinfacht lässt sich sagen, dass die exklusive Kanonisierung angewandt werden soll, wenn die Signatur aus dem Kontext herausgenommen werden und dabei ihre Gültigkeit erhalten bleiben soll.

Konsequenz und wichtig: Zur Angabe der Elemente und deren Attribute **dürfen keine DTD Angaben** in den XML-Objekten gemacht werden, weil diese bei der Kanonisierung zerstört werden. Es müssen XML-Schemas dazu verwendet werden, ansonsten fliessen diese Angaben zur Struktur nicht in die XML-Signatur ein und sind folglich nicht bezüglich Authentizität und Integrität geschützt.

5.2.4 Algorithmen für die Prüfsummen der Objekte

In den Standards [RFC 3275] und [RFC 4051] sind Algorithmen (Hashfunktionen) für den Message Digest (kryptographische Prüfsumme) aufgeführt. Hier wird wegen der Sicherheit präzisiert, welche und wie diese anzuwenden sind.

MUST: Im Bereich der Signatur dürfen nur SHA-x und RIPEMD5 angewandt werden.

SHOULD: Kryptographische Prüfsummen mit einer Mindestlänge von 256 Bit sollten für die rechtlich verbindliche Signatur eingesetzt werden.

MUST NOT: MD5 darf nicht verwendet werden.

Grund: MD5 gilt heute im Zusammenhang mit Signaturen als zu unsicher.

MUST: Für alle in die Signatur aufzunehmenden Dokumente muss der gleiche Algorithmus für den Message Digest eingesetzt werden.

Grund: Vereinfacht ausgedrückt, die kryptographische Prüfsummenbildung zur Bildung der Signatur ist so stark, wie der schwächste eingesetzte Algorithmus zur Bildung der Message Digest.

5.2.5 Kanonisierung (engl. Canonicalization) der Prüfsummenelemente

Die Hashwerte der einzelnen Dokumente und weitere Angaben dazu werden zuerst aufbereitet (u.a. serialisiert), s. auch Anhang A (vereinfachte bildliche Darstellung der XML-Signatur). Dann wird das Resultat signiert, die Aufbereitung der Daten wird aber nicht gespeichert.

Wichtig ist es deshalb, dass bei der Verifikation der Signatur die gleiche Aufbereitung angewandt wird, wie bei der Herstellung. Ansonsten wird die Verifikation der Signatur ein ungültiges (fehlerhaftes) Ergebnis liefern.

MUST: Die Methoden der Kanonisierung, welche in den Standards [RFC 3076] und [RFC 3741] angegeben sind, sind zu unterstützen.

MUST NOT: Andere Verfahren zur Kanonisierung dürfen aus Sicherheitsüberlegungen nicht angewandt werden.

Grund: s. Kapitel 8.1 im Standard [RFC 3275] und Kapitel 5.2.3 „Transformation der Objekte“.

5.2.6 Verfahren für Signatur

5.2.6.1 Hash Algorithmen

MUST: Zur Bildung der Prüfsumme für die Signatur (über die Liste der Prüfsummen der einzelnen Objekte) muss das gleiche Verfahren wie zur Bildung der Prüfsummen (Message Digest) der einzelnen Objekte verwendet werden.

Grund: Vereinfacht ausgedrückt, die kryptographische Prüfsummenbildung ist so stark, wie der schwächste eingesetzte Algorithmus.

5.2.6.2 Asymmetrische Verfahren

Hier wird definiert, welche asymmetrischen Verfahren zur Bildung der Signatur eingesetzt werden sollen.

MUST: Im Unterschied zum Standard [RFC 3275] **muss** RSA unterstützt werden.

Grund: In der Schweiz werden (fast) ausschliesslich RSA Schlüssel in die qualifizierten Zertifikate eingefügt und in Chipkarten verteilt. Zudem wird bei der Verschlüsselung zum Transport des symmetrischen Schlüssels im W3C Standard nur RSA empfohlen.

Es macht zudem keinen Sinn, zwei verschiedene asymmetrische Verfahren im Bereich der Signatur und zum Schutz der Vertraulichkeit von Dokumenten einzusetzen.

MAY: Aus diesen Gründen ist die Verwendung des DSA Algorithmus im Unterschied zum Standard [RFC 3275] optional und sollte eher nicht eingesetzt werden.

MUST: Falls Hashwerte mit der Länge grösser als 256 Bit eingesetzt werden und das DSA Verfahren verwendet wird, dann müssen die Verfahren verwendet werden, welche im Kapitel 2.3.6 von [RFC 4051] aufgeführt sind. Dabei ist zu beachten, dass die möglichen Werte im asymmetrischen Verfahren mindestens gleich gross sind wie die Anzahl der möglichen Hashwerte.

Grund: Bereits bei der Bildung der Signatur können sonst Kollisionen entstehen.

5.2.6.3 HMAC

In den Standards [RFC 3275] und [RFC 4051] sind Algorithmen (Hashfunktionen) für die Authentizität aufgeführt. Die Sicherung der Authentizität ist gemäss Standard auch mittels HMAC erlaubt.

MUST NOT: Das HMAC Verfahren darf bei nicht voll automatisierten Prozessen nicht eingesetzt werden.

Grund: Das Schlüsselmanagement sollte vom Benutzer aus Gründen der sicheren Ablage dieser Schlüssel ferngehalten werden.

MUST: Der Schlüssel zum HMAC muss mindestens eine Länge von 128 Bit aufweisen.

Grund: Weniger lange zufällig erzeugte Schlüssel gelten heutzutage als eher unsicher.

5.2.7 Angaben zum Unterzeichnenden

Gemäss Standard können Angaben zum Unterzeichnenden, dessen Schlüssel für die Signaturverifikation und zu dessen Zertifikat gemacht werden.

MUST: Falls Angaben zum Unterzeichnenden und dessen öffentlichen Schlüssel gemacht werden, sind nur Angaben zu verwenden, welche auch im X.509 Zertifikat enthalten sind.

Grund: Angaben in Zertifikaten sind verlässlich, s. zudem auch [RFC 3850] oder Kapitel 11 [Mud].

Andere Zertifikatsformen wie PGP oder SPKI Zertifikate werden in der Schweiz kaum verwendet und sind übrigens für die der Handunterschrift gleichgestellten elektronischen Signaturen auch nicht erlaubt.

5.2.7.1 Anzeige an den Benutzer

Applikationen können dem Benutzer, welche die Signatur prüft, die zusätzlichen Angaben zum Unterzeichner anzeigen.

MUST: Werden Angaben zur Signierentität (wie Person, Institution oder Dienst) dem Empfänger angezeigt, dann muss die SW verifizieren, ob die gemachten Angaben mit den Informationen im Zertifikat übereinstimmen, welches für die Verifikation der Signatur verwendet werden soll. Stimmen die beiden Angaben nicht überein, dann muss bei der Verifikation der Signatur mittels einer Warnmeldung darauf hingewiesen werden.

SHOULD: Die Signatur sollte bei unterschiedlichen Angaben nicht akzeptiert werden.

Grund: S. [RFC 3850] oder Kapitel 11 [Mud]

5.2.8 Angaben zu den Unterobjekten

Gemäss Standard [RFC 3275] ist es erlaubt, Angaben zum Typ (MIME Type) der zu signierenden Objekte beizufügen.

SHOULD: Angaben zum Typ der zu signierenden Objekte sollten beigefügt werden.

Grund: Angaben zum Objekttyp erleichtern unter Umständen die Komposition (Zusammenstellung) des Dokuments.

MUST: Werden Angaben gemacht, dann muss der entsprechende Standard [RFC 3023] eingehalten werden.

Grund: Nicht standardisierte Angaben zum Typ des Objekts erschweren die Interoperabilität und folglich die Komposition des Objekts.

5.2.9 Schlüsselvereinbarung/-einigung (Key Agreement)

Es besteht gemäss XML-Signatur Standard [RFC 3275] auch noch die Möglichkeit, mit der XML Signatur Struktur Schlüssel auszutauschen oder sich auf einen Schlüssel zu einigen.

SHOULD NOT: Key Agreement (Schlüsselvereinbarung)

Grund: Hierzu gibt es bessere und vor allem standardisierte Technologien wie SSL/TLS als ein XML-Objekt einzusetzen.

MUST: Falls doch eine Schlüsselvereinbarung mit einem XML-Objekt durchgeführt wird, dann muss sie vollautomatisch sein, so dass sich gegebenenfalls ein Benutzer nicht um das Schlüsselmanagement kümmern muss.

Grund: Das Schlüsselmanagement sollte vom Benutzer aus Gründen der sicheren Ablage ferngehalten werden.

5.3 Verschlüsselung

5.3.1 Grundlegendes

Der XML Encryption Standard erlaubt es, ganze Dokumente oder nur Teile davon zu verschlüsseln, dies bei der Signatur, s. W3C Standard [Decryption Transforms for XML-Signature]. Der W3C Standard zur XML-Verschlüsselung erlaubt die folgenden 3 Arten von XML-Verschlüsselungen:

- Die Verschlüsselung des Inhalts eines XML-Elements
- Die Verschlüsselung des XML-Elements und dessen Inhalt
- Die Verschlüsselung von irgendwelchen Objekten, welche auch XML-Bestandteile haben können.

MUST: Alle sicherheitsrelevanten (vertrauliche) Informationen des Verwaltungsdokuments sind zu verschlüsseln.

Grund: Liegen sicherheitsrelevante (vertrauliche) Teile des Dokuments im Klartext vor, so können sensitive Daten offengelegt werden und somit in falsche Hände geraten.

SHOULD: Informationen, welche Rückschlüsse auf den verwendeten Verschlüsselungsschlüssel oder den verschlüsselten Text erlauben, sollten ebenfalls verschlüsselt werden.

Grund: Die Prüfsummenwerte oder Signaturen sollten nicht im Klartext vorliegen. Dies erlaubt unter anderem eine beschleunigte Brute Force Attacke auf die Verschlüsselung, weil eine beschleunigte Plausibilitätsprüfung für einen Kandidaten eines Entschlüsselungsschlüssels durchgeführt werden kann, s. dazu [Mud] Seite S. 204.

Die Angaben zur Struktur des verschlüsselten Objekts (wie MIME oder JPEG) sollten ebenfalls verschlüsselt werden, weil dies ebenfalls Rückschlüsse auf den Verschlüsselungsschlüssel ermöglichen kann.

5.3.2 Angaben zu den Unterobjekten

Gemäss W3C Standard zur XML-Verschlüsselung ist es erlaubt, Angaben zum Objekttyp (MIME Type) der zu verschlüsselnden Dateien im Klartext beizufügen.

MUST NOT: Nähere Angaben im Klartext zum Typ (MIME Type) der zu verschlüsselnden Unterobjekten dürfen nicht im Klartext gemacht werden. Falls Angaben zu den Objekten in den Elementen vorhanden sind, sind diese auch zu verschlüsseln.

Ausnahme der soeben gemachten Empfehlung: Falls es sich beim verschlüsselten Objekt um ein komprimiertes Objekt handelt, dann muss der MIME Type im Klartext angegeben werden.

Grund: Etwelche Informationen zu den verschlüsselten Objekten sind zu unterlassen, weil dies unter Umständen die Sicherheit der Verschlüsselung und somit der Vertraulichkeit beeinträchtigen kann.

5.3.3 Aufbereitung der zu verschlüsselnden Daten

Das ganze Verwaltungsdokument ist gegebenenfalls entsprechend so aufzubereiten, dass es unabhängig vom Kontext ist, worin es sich vorher befunden hat. Ansonsten besteht die Gefahr, dass Unterobjekte ungewollt verschlüsselt abgelegt werden und für andere Berechtigte nicht mehr zugänglich sind. Entsprechend sind auch die Verweise dann im XML-Objekt anzupassen. Falls das Verwaltungsdokument auch noch vor der Verschlüsselung signiert werden sollte, dann ist die Signatur erst dann einzusetzen, wenn das Verwaltungsdokument vom Kontext unabhängig ist, ansonsten verliert die Signatur ihre Gültigkeit.

Folgende Verfahren zur Verschlüsselung der XML-Objekte werden empfohlen

- XML-Verschlüsselung, falls das zu verschlüsselnde Hauptobjekt aus einem Hauptobjekt besteht.
- Alle Unterobjekte und das Hauptobjekt werden in ein ZIP File eingefügt. Dieses wird dann verschlüsselt. Man kann das ZIP File direkt mittels PKCS#7 oder neuer IETF Standard CMS (Cryptographic Message Syntax [RFC 3852]).
- Das Objekt nach Base64 codieren, in eine XML-Datei einfügen und danach den entsprechenden Objektteil verschlüsseln.

Weniger empfohlen wird, bei der Verschlüsselung mit Cipher Reference zu arbeiten, s. Kapitel 3.3.1 des WC3 Standards [XML Encryption].

SHOULD NOT: Signierte Objekte sollten nie verändert werden. Müssen aber die signierten Objekte aus irgendwelchen Gründen vor der Verschlüsselung doch transformiert werden, dann muss gelten:

MUST: Falls eine Kanonisierung auf die zu verschlüsselnden Daten angewandt wird, dann muss bei der XML-Verschlüsselung die gleiche Methode zur Kanonisierung wie bei der Bildung der Signatur angewandt werden.

Grund: Im Unterschied zur Signatur verändert die Aufbereitung (die Kanonisierung) der Daten vor der Verschlüsselung das Ursprungsobjekt. Deshalb muss darauf geachtet werden, dass durch diese Aufbereitung der Daten die bereits geleistete Signatur über das unverschlüsselte Dokument nicht zerstört wird.

Eventuell könnte man XML-Dokumente, welche sowohl signiert und verschlüsselt werden sollten, gegebenenfalls zuerst für die Verschlüsselung aufbereiten, dann signieren und dann erst verschlüsseln. Zur Problematik der Kanonisierung im Zusammenhang mit der Signatur sind weitere Informationen bei http://www.w3.org/Security/ aufgeführt.

Beispiel für Inkompatibilität und Standard konformer Anwendung: Bei der Kanonisierung zur späteren Bildung Signatur wird eine darin empfohlene Methode angewandt, welche die Kommentare in den XML-Objekten unverändert lässt. Bei der Verschlüsselung wird eine vom W3C empfohlene Kanonisierung angewandt, welche die Kommentare entfernt. Konsequenz ist, dass die Signatur nach der Verschlüsselung nicht mehr gültig ist.

5.3.4 Algorithmen für Verschlüsselung

Grundsätzlich sind alle im W3C Standard aufgeführten Algorithmen zur XML-Verschlüsselung zu unterstützen.

5.3.5 Schlüsselvereinbarung/-einigung (Key Agreement)

SHOULD NOT: Key Agreement (Schlüsselvereinbarung)

Grund: Hierzu gibt es bessere und standardisierte Technologien wie SSL/TLS als einen XML-Objekt einzusetzen.

MUST: Falls doch eine Schlüsselvereinbarung über ein XML-Objekt durchgeführt wird, dann muss sie vollautomatisch sein, so dass sich gegebenenfalls ein Benutzer nicht um das Schlüsselmanagement kümmern muss.

Grund: Das Schlüsselmanagement sollte vom Benutzer aus Gründen der sicheren Ablage der Schlüssel ferngehalten werden.

6 Alternativlösungen

Wie im vorhergehenden Kapitel ersichtlich, kann die XML Security nicht völlig losgelöst von der XML Applikation betrachtet werden, siehe auch Kapitel 3 und 4 [SOAP Security with Attachements]. Dies bedingt, dass die Security auf die Applikation abgestimmt sein muss und umgekehrt, was wiederum zu erheblichen Schwierigkeiten in den praktischen Anwendungen und Umsetzungen führen kann.

Um den Problemen rund um die (qualifizierte) XML-Signatur auszuweichen, wird bei INCA-Mail der Schweizerischen Post, das XML- in ein PDF-Dokument umgewandelt und dann signiert. PDF in den älteren Versionen und PDF/A haben den Vorteil, dass der Benutzer sehen kann, welche Signatur er prüft (Art. 6 Abs. 3 lit. a ZertES), die Daten bei Bedarf anzeigen (Art. 6 Abs. 3 lit. c ZertES) und sicherheitsrelevante Veränderungen feststellen kann (Art. 6 Abs. 3 lit. g ZertES).

Der letztgenannte Ansatz hat den Nachteil, dass wohl das PDF-Objekt signiert und somit authentisiert und bezüglich Integrität geschützt ist, aber die XML-Dateien nicht. Der Schutz der XML-Objekte wird aber gegebenenfalls zwecks Weiterverarbeitung des Inhalts benötigt. Ein Lösungsansatz hierzu wäre, alles, was zum XML-Dokument gehört und dafür relevant ist, neutral zum Kontext oder zur Umgebung abzuspeichern. Die internen Verweise müssen gegebenenfalls angepasst werden, falls sie im ursprünglichen Dokument nicht relativ sind, sondern absolut aufgeführt werden. Das Ganze wird danach komprimiert (gezippt) und in ein ZIP-File gepackt. Dieses File wird dann signiert und falls erforderlich auch noch verschlüsselt gemäss dem entsprechenden RFC Standard [RFC 3852]. Der Nachteil bei der Signatur des ZIP-Files im Benutzerumfeld ist, dass der Benutzer nicht genau anzeigen lassen kann, was er signiert hat, nämlich das ZIP-File. Dem Benutzer wird nämlich nicht das ZIP-File angezeigt, sondern dessen dekomprimierten Inhalt.

Das Komprimieren des Files verhindert auch nicht, dass XML-Verwaltungsdokumente erzeugt werden können, welche bei gewissen Applikationen in der Ansicht identisch sind, aber einen unterschiedlichen Hashwert aufweisen.

7 Haftungsausschluss/Hinweise auf Rechte Dritter

eCH-Standards, welche der Verein **eCH** dem Benutzer zur unentgeltlichen Nutzung zur Verfügung stellt, oder welche **eCH** referenziert, haben nur den Status von Empfehlungen. Der Verein **eCH** haftet in keinem Fall für Entscheidungen oder Massnahmen, welche der Benutzer auf Grund dieser Dokumente trifft und / oder ergreift. Der Benutzer ist verpflichtet, die Dokumente vor deren Nutzung selbst zu überprüfen und sich gegebenenfalls beraten zu lassen. **eCH**-Standards können und sollen die technische, organisatorische oder juristische Beratung im konkreten Einzelfall nicht ersetzen.

In **eCH**-Standards referenzierte Dokumente, Verfahren, Methoden, Produkte und Standards sind unter Umständen markenrechtlich, urheberrechtlich oder patentrechtlich geschützt. Es liegt in der ausschliesslichen Verantwortlichkeit des Benutzers, sich die allenfalls erforderlichen Rechte bei den jeweils berechtigten Personen und/oder Organisationen zu beschaffen.

Obwohl der Verein **eCH** all seine Sorgfalt darauf verwendet, die **eCH**-Standards sorgfältig auszuarbeiten, kann keine Zusicherung oder Garantie auf Aktualität, Vollständigkeit, Richtigkeit bzw. Fehlerfreiheit der zur Verfügung gestellten Informationen und Dokumente gegeben werden. Der Inhalt von **eCH**-Standards kann jederzeit und ohne Ankündigung geändert werden.

Jede Haftung für Schäden, welche dem Benutzer aus dem Gebrauch der **eCH**-Standards entstehen ist, soweit gesetzlich zulässig, wegbedungen.

8 Urheberrechte

Wer **eCH**-Standards erarbeitet, behält das geistige Eigentum an diesen. Allerdings verpflichtet sich der Erarbeitende sein betreffendes geistiges Eigentum oder seine Rechte an geistigem Eigentum anderer, sofern möglich, den jeweiligen Fachgruppen und dem Verein **eCH** kostenlos zur uneingeschränkten Nutzung und Weiterentwicklung im Rahmen des Vereinszweckes zur Verfügung zu stellen.

Die von den Fachgruppen erarbeiteten Standards können unter Nennung der jeweiligen Urheber von **eCH** unentgeltlich und uneingeschränkt genutzt, weiterverbreitet und weiterentwickelt werden.

eCH-Standards sind vollständig dokumentiert und frei von lizenz- und/oder patentrechtlichen Einschränkungen. Die dazugehörige Dokumentation kann unentgeltlich bezogen werden.

Diese Bestimmungen gelten ausschliesslich für die von **eCH** erarbeiteten Standards, nicht jedoch für Standards oder Produkte Dritter, auf welche in den **eCH**-Standards Bezug genommen wird. Die Standards enthalten die entsprechenden Hinweise auf die Rechte Dritter.

Anhang A – Überblick XML-Signatur Bildung

In folgender Grafik ist sehr vereinfacht dargestellt, wie eine XML-Signatur hergestellt wird:

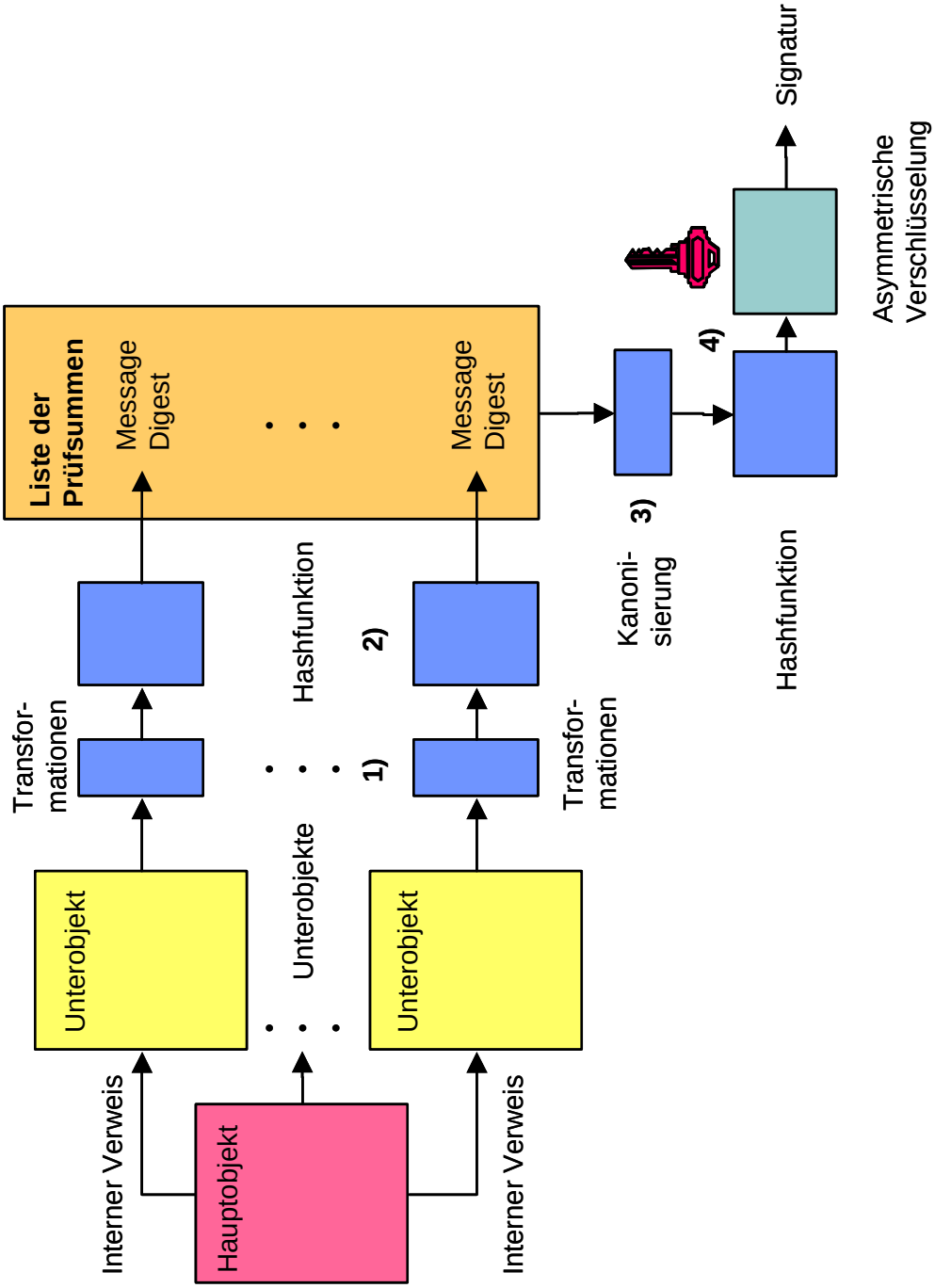


Abb. A-1 Signaturbildung

Die Herstellung der Signatur erfolgt in 4 Schritten:

1. Sämtliche der zu signierenden Objekte werden einer oder mehrerer Transformationen unterzogen, wobei gemäss Standard [RFC 3275] die Transformationen für die jeweiligen Objekte jeweils unterschiedlich sein können.
2. Von allen transformierten Objekten wird je eine kryptographische Prüfsumme (Message Digest) hergestellt, wobei gemäss Standard [RFC 3275] das Verfahren zur Herstellung der Prüfsumme für die jeweiligen Objekte unterschiedlich gewählt werden kann. In diesem Dokument wird aber empfohlen, jeweils das gleiche Verfahren einzusetzen.
3. Die Liste der Message Digest (Hashwerte oder Prüfsummen) wird dann kanonisiert.
4. Von der Kanonisierung wird dann die Signatur hergestellt. Gemäss Standard [RFC 3275] dürfte hier ein anderes Verfahren zur Herstellung der Prüfsumme für die Signatur als im vorangegangenen Schritt 2 eingesetzt werden. In diesem Dokument wird aber empfohlen, pro Signatur immer das gleiche Verfahren zur Herstellung einer Prüfsumme einzusetzen.

Anmerkung: Von jedem Objekt, wovon eine Prüfsumme hergestellt wird, werden zusätzliche Angaben gemacht und diese in der Liste so aufgeführt, dass sie dem Objekt zugeordnet werden können, wie:

- Referenz (Verweis) auf das Objekt, von welchem die Prüfsumme hergestellt worden ist.
- Angabe zu den Transformationsverfahren
- Angaben zum Typ des Objekts
- Angabe zum Algorithmus, den man zur Herstellung der Prüfsumme verwendet hat

Bei der Signatur können zusätzliche Angaben gemacht werden, wie

- Angaben zum Zertifikat, welches zur Verifikation der Signatur benötigt werden.
- Angaben zur Entität, welche die Signatur hergestellt hat.

Anhang B – Referenzen & Bibliographie

Fachliteratur

[MOV] Alfred Menezes, Paul van Orschot, Vanstone Scott, Handbook of Applied Cryptography, CRC Press 1996, ISBN 0 8493 8523 7 http://cacr.math.uwaterloo.ca/hac/

[Mud] Muster Daniel, Digitale Unterschriften und PKI, 3. Auflage 2006, ISBN 3 9522387 3 2

[Nem] Mark O'Neal, et al, Web Services Security, Mc Graw Hill/ Osborne, 2003, ISBN 0 07 222471 1

[Sch] Schneier Bruce, Angewandte Kryptographie, Addison Wesley, 1. Auflage 1996, ISBN 3 89319 854 7

eCH (www.ech.ch)

eCH-0018 XML Best Practices

eCH-0036 Dokumentation für den XML-orientierten Datenaustausch

IETF Standards (www.ietf.org)

RFC 2315 PKCS #7: Cryptographic Message Syntax Version 1.5

RFC 3023 XML Media Types

RFC 3076 Canonical XML

RFC 3275 XML Signature Syntax and Processing

RFC 3741 Exclusive XML Canonicalization

RFC 3850 S/MIME v. 3.1 Certificate Handling

RFC 3852 Cryptographic Message Syntax (CMS)

RFC 4051 Additional XML Security Unique Resource Identifiers

W3C Standards (www.w3C.org)

Decryption Transforms for XML Signature Recommendation, December 2002

XML Encryption and Syntax Processing Recommendation, December 2002

XML Signature and Syntax Processing Recommendation, February 2002

CEN Standards

CWA 14170: CEN (European Committee for Standardization), Security Requirements for Signature Creation Applications, May 2004

CWA 14171 CEN (European Committee for Standardization), General Guidelines for electronic signature verification, May 2004

OASIS Standards (www.oasis-open.org)

Web Services Security, SOAP Messages Security 1.1, February 2006

Web Services Security, SOAP Messages with Attachments (SwA) Profile 1.1, February 2006

Gesetzestexte

ZertES: Bundesgesetz vom 19. Dezember 2003 über Zertifizierungsdienste im Bereich der elektronischen Signatur (SR 943.03)

Anhang C – Mitarbeit & Überprüfung

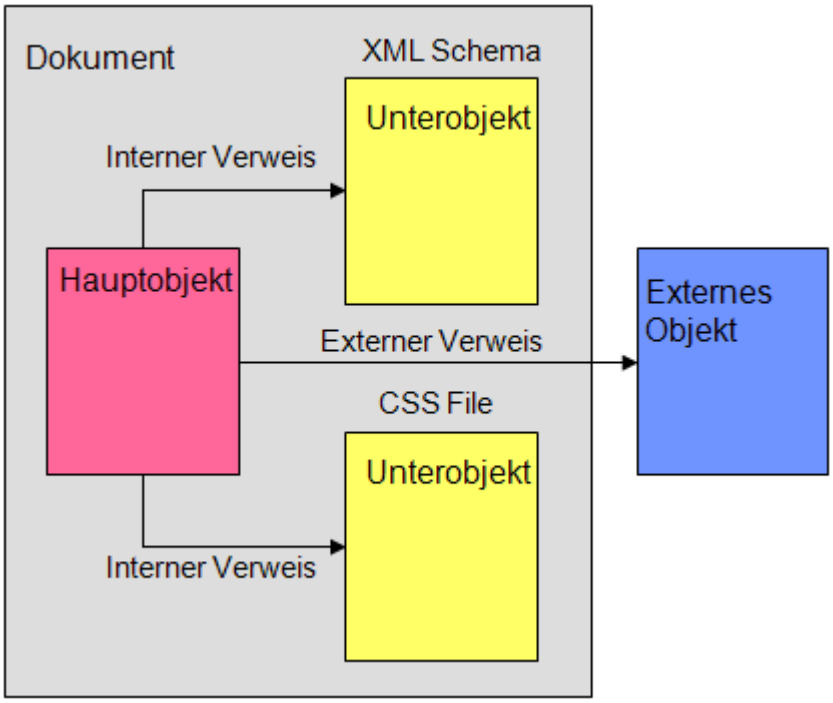
vgl. Autoren in Titelblatt

Anhang D – Glossar

Dokument	Das Dokument besteht aus dem Hauptobjekt und aus den dazu gehörigen (darauf *intern* verwiesenen) Unterobjekten. Ein XML-Dokument ist ein Dokument, dessen Hauptobjekt eine XML-Struktur aufweist.
Hauptobjekt	Ursprungsobjekt, von welcher die expliziten Verweise starten. Objekt, welches das Wurzelelement enthält.
Kanonisierung	Das gleiche XML-Objekt kann sich bei unterschiedlichen Betriebssystemen unterschiedlich präsentieren. Mit der Signatur will man aber nicht nur die Herkunft des Objektes bestimmen, sondern auch die Integrität (die Möglichkeit, eine Veränderung festzustellen) des Objektes schützen. Um zu verhindern, dass das gleiche XML-Objekt sich auf unterschiedlichen Betriebssystemen unterschiedlich präsentiert, wendet man eine Kanonisierung auf das XML-Objekt an. Nach der Kanonisierung setzt sich dieses Objekt auf den verschiedenen Systemen Byte für Byte gleich zusammen. Standards zur Kanonisierung s. [RFC 3076] und [RFC 3741].
Qualifizierte Signatur	S. Art. 2 Bst. c ZertES: Eine fortgeschrittene elektronische Signatur, die auf einer sicheren Signaturerstellungseinheit und auf einem qualifizierten und zum Zeitpunkt der Erzeugung gültigen Zertifikat beruht.
Qualifiziertes Zertifikat	Ein digitales Zertifikat, das die Anforderungen des Artikels 7 ZertES erfüllt.
Signaturtypen	Es sind folgende Signaturtypen standardisiert und in der Technik am weitesten verbreitet: ▪ PKCS#7 Signature s. [RFC 2315], neuer Standard bei IETF CMS (Cryptographic Message Syntax [RFC 3852] ▪ XML Signature s. [RFC 3275] Das technische Prinzip bei beiden Signaturen ist prinzipiell gleich. Sie unterscheiden sich aber darin, welche zusätzlichen Informationen der Signatur beigelegt werden und wie sie strukturiert sind.
Transformation	Transformation im Sinne des Standards [RFC 3275] bedeutet, dass das

	Objekt in der gewünschten Art gefiltert, verarbeitet oder kanonisiert wird, bevor die kryptographische Prüfsumme (Hashwert) über das durch die Transformation modifizierte Objekt berechnet wird.
Unterobjekt	Objekt, auf welches *intern* verwiesen wird, wie z.B. eine CSS-File oder eine Bilddatei.
Verwaltungsdokument	Ein Dokument, welches die an einem Verwaltungsprozess beteiligten Akteure einander zusenden, um einen bestimmten Geschäftsfall auszulösen, zu protokollieren, zu bearbeiten oder zu erledigen. Zur Gewährleistung der Nachvollziehbarkeit sind sie zu archivieren. Beispiele sind: Ausgefüllte Antragsformulare, Erlasse, Evaluationen, Berichte, Registerauszüge etc.

Verweis	Mit einem Verweis in einem Objekt wird entweder *implizit* oder *explizit* auf ein anderes Objekt oder auf eine Steueranweisung referenziert. Ein implizierter Verweis referenziert auf ein nicht existierendes Objekt oder eine nicht ausgeschriebene Steueranweisung, sondern auf etwas, was innerhalb der Kommunikationsgesellschaft zuvor vereinbart worden ist. Ein existierender Verweis auf eine Steueranweisung, resp. auf ein Objekt ist z.B. ein Verweis auf einen Code in JavaScript oder ein XML-Schema. Ein implizierter Verweis ist z.B. die Angabe: <Webseite xmlns:html="http://www.w3.org/TR/REC-html40"> Jedes Element innerhalb des Elements „Webseite", welches mit <html: > beginnt wird als Information in HTML interpretiert. Bei den expliziten Verweisen wird zwischen Dokument internen und externen Verweisen unterschieden. Dokument interne Verweise sind Verweise auf Unterobjekte (z.B. Dateien), welche Bestandteil des Dokuments sind, wie Schemas oder Bilder. Externe Verweise sind jedoch Verweise auf externe Objekte (Haupt- oder Unterobjekte), welche Zusatzinformationen enthalten, aber für die Interpretation des Dokuments nicht wesentlich sind. Beispiel eines externen Verweises ist eine Quellenangabe, welche z.B. in HTML oder XML geschrieben ist. Hierzu folgende Illustration:



Anmerkung: Was nun als einen externen oder internen Verweis betrachtet wird, hängt einerseits von den Intentionen des Verfassers des XML-Dokuments, von der Kommunikationsgemeinschaft oder von der Kommunikationsplattform ab.

XML-Dokument	Ein Dokument, dessen Hauptobjekt eine XML-Struktur aufweist.
XML-Verwaltungsdokument	Ein Verwaltungsdokument, dessen Hauptobjekt eine XML-Struktur aufweist.

Anhang E – Abkürzungen

CSS	Cascading Style Sheets Language
DTD	Document Type Definition
GRDDL	Gleaning Resource Descriptions from Dialects of Languages
HMAC	Keyed-Hash Message Authentication Code
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IRI	Internationalized Resource Identifier
RDDL	Resource Directory Description Language
RDF	Resource Description Framework
resp.	respektive
RSA	Rivest Shamir Adleman Public Key Verfahren
Rz	Randziffer
SOAP	Service Oriented Application Protocol
SSL	Secure Socket Layer
TLS	Transport Layer Security
URI	Universal Resource Identifier
W3C	World Wide Web Consortium
XHTML	Extensible Hypertext Markup Language
XLink	Extensible Linking Language
XML	Extensible Markup Language
XSLT	XSL Transformations