eCH-0091 – Standard zu XML-Signatur und -Verschlüsselung

Name	Standard zu XML-Signatur und -Verschlüsselung
eCH-Nummer	eCH-0091
Kategorie	Standard
Reifegrad	Definiert
Version	2.0.0
Status	Genehmigt
Beschluss am	2021-03-02
Ausgabedatum	2021-03-10
Ersetzt Version	1.0 – Major Change
Voraussetzungen	-
Beilagen	-
Sprachen	Deutsch (Original), Französisch (Übersetzung)
Autoren	Fachgruppe SAGA Büchler Georg (KOST), Müller Adrian (SwissSign AG), Muster Daniel (it-rm IT Riskmanagement GmbH) Niederberger Marcel (ESTV) von Niederhäuser Michael (Bit) Rötzer Hubert Schmid Josef Waldegger Hans-Peter (Swisscom) Fachgruppe XML für Version 1.0 Daniel Muster (Initiant dieses Themas) Willy Müller 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 u.a. 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 weist einerseits auf die damit verbundenen Probleme hin, 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

Hinweis

Aus Gründen der besseren Lesbarkeit und Verständlichkeit wird im vorliegenden Dokument bei der Bezeichnung von Personen ausschliesslich die maskuline Form verwendet. Diese Formulierung schliesst Frauen in ihrer jeweiligen Funktion ausdrücklich mit ein.

1 Einleitung

1.1 Status

Genehmigt: Das Dokument wurde vom Expertenausschuss genehmigt. Es hat für das definierte Einsatzgebiet im festgelegten Gültigkeitsbereich normative Kraft.

1.2 Ziel des Dokuments

Dieses Dokument weist auf (Sicherheits)probleme hin, welche bei der Verschlüsselung und Signierung von XML-Dokumenten und XML-Objekten auftreten können, und unterbreitet Lösungsvorschläge zur Behebung der besagten Probleme; 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 sollen; dies zum Beispiel im Unterschied zu einer SOAP oder SAML Meldung Diese sind eine von vielen XML-Anwendungen.

Bei der Signatur und Verschlüsselung von XML-Verwaltungsdokumenten und XML-Objekten 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 Bestandteile der Signatur unverändert und die Signatur innerhalb einer vorgesehenen Zeit gültig und prüfbar bleiben.

In den folgenden Kapiteln liegt der Fokus auf XML-Verwaltungsdokumenten, wobei auch Empfehlungen zu XML-Anwendungen in der Datenkommunikation abgegeben werden:

- Kapitel 2 «Erläuterung der Problematik»
- Kapitel 3 «Risiken und Massnahmen»
- Kapitel 4 «Präzisierung der bestehenden Standards»
- Kapitel 5 «Alternativen»

- Anhang A «Überblick XML-Signatur Bildung»

1.3 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», means 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.)

1.4 Auswahl der Standards

Im Umfeld des nach ZertES anerkannten Zertifizierungsdienstanbieters sind prioritär Europäische Standards massgebend. Für den hier vorliegenden Fall existiert jedoch kein ETSI-Standard. Deswegen basieren die hier vorliegenden Empfehlungen auf den W3C-Standards für XML-Signatur und -Verschlüsselung. Für die langfristige Gültigkeit elektronischer XML-Signaturen sind entsprechende XML-Standards (XAeDS-Serie) von ETSI erarbeitet und publiziert worden, wobei die erwähnten W3C-Standards als Grundlage für die weiteren Ausführungen zur langfristen Gültigkeit der XML-Signaturen in den erwähnten ETSI-Standards dienen.

2 Erläuterung der Problematik

Im Standard [CWA 14170], resp. [CWA 14171] wird dargelegt, welche Risiken beim Erstellen, resp. beim Verifizieren einer elektronischen Signatur bestehen und welche Sicherheitsmassnahmen dagegen getroffen werden können/sollen. Dies erfolgt unabhängig von dem zugrundeliegenden Signaturformat. Für spezifische Signaturformate wie CMS oder XML sollen primär die entsprechenden Standards von ETSI konsultiert werden, wie TS 119 102-1 V1.2.1.

Das hier vorliegende Dokument im Gegensatz zu den erwähnten CWA Standards beschränkt sich u.a. 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.

2.1 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 von der Signatur erfasst, 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 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.

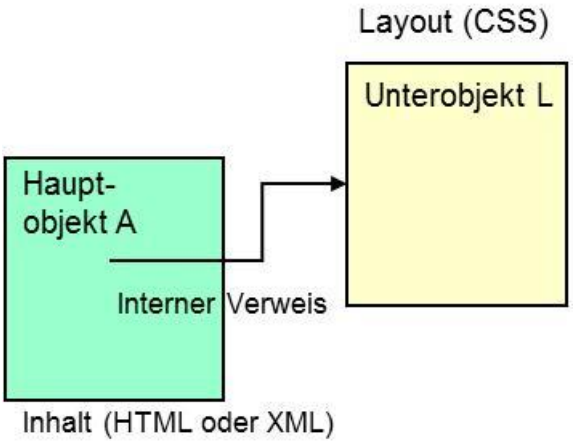


Abbildung 1 Trennung von Inhalt und Darstellung bei einem Dokument

Carl hat die Layout-Information im Unterobjekt L (z.B. im CSS Format) so definiert, dass gewisse Passagen mit weisser Schrift auf weissem Hintergrund präsentiert werden. Er unterbreitet das Dokument mit der Layout-Information 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.

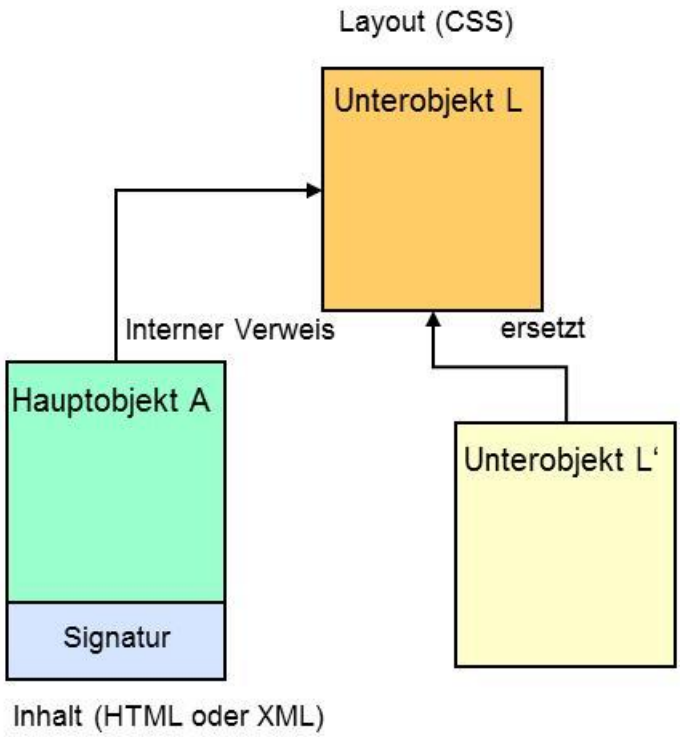


Abbildung 2 Auswechslung von Inhalten, die Gültigkeit der Signatur bleibt erhalten

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 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.

2.2 Probleme bei der Verschlüsselung

Probleme treten bei der XML-Verschlüsselung gemäss W3 Standard eines (Verwaltung)Dokuments in XML auf: Sei, 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 soll 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.

2.3 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.

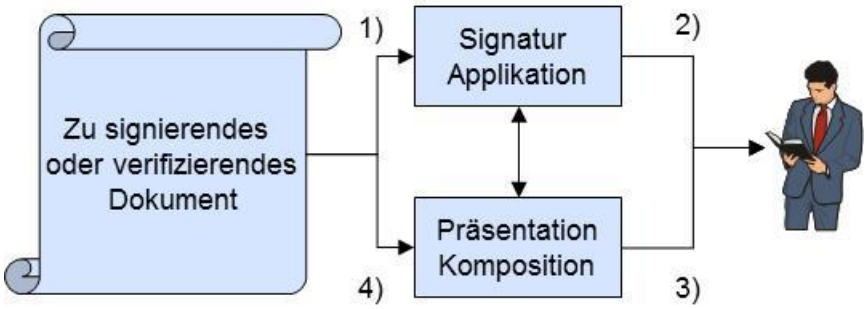


Abbildung 3 Modell für die Erstellung und Verifikation der Signatur

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, soll 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.

3 Risiken und Massnahmen

Die Anwendungsfälle zu den hier nun untersuchten Risiken werden unterteilt in

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

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

3.1 Einleitende Informationen zur XML-Signatur

Der XML-Signatur Standard W3C-Sig 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).

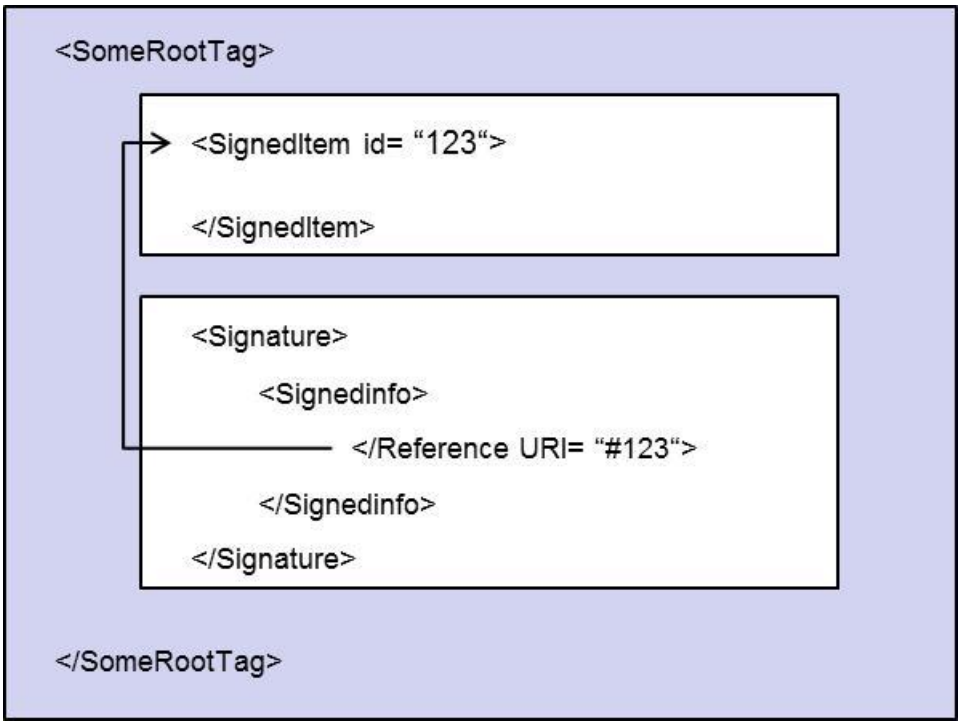


Abbildung 4 Detached Signature (erste Ausprägung)

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

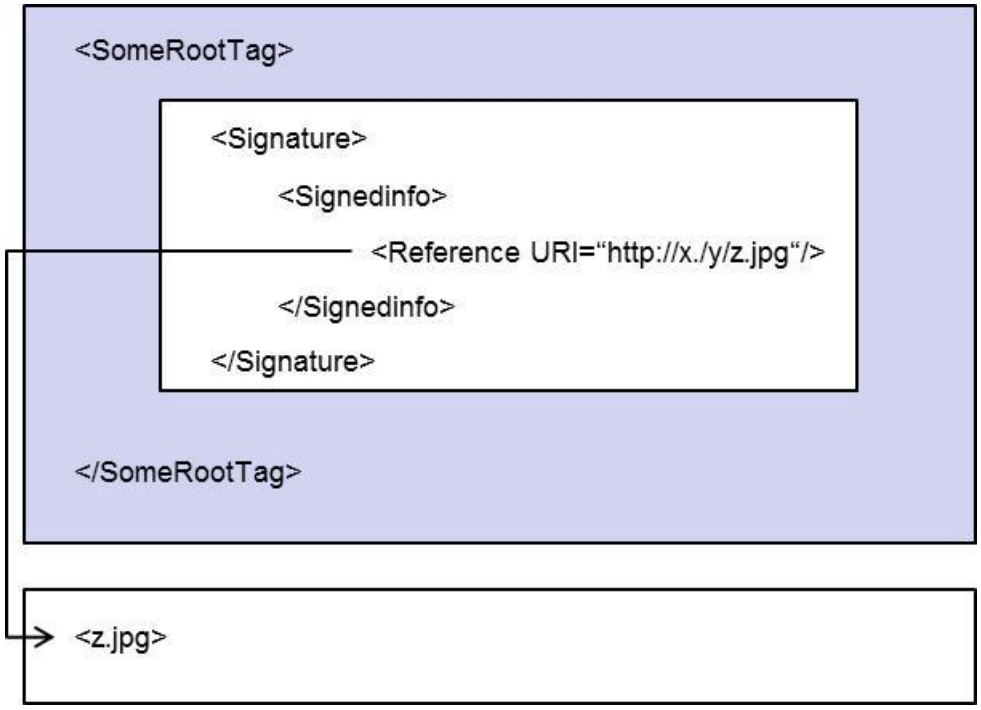


Abbildung 5 Detached Signature (zweite Ausprägung)

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

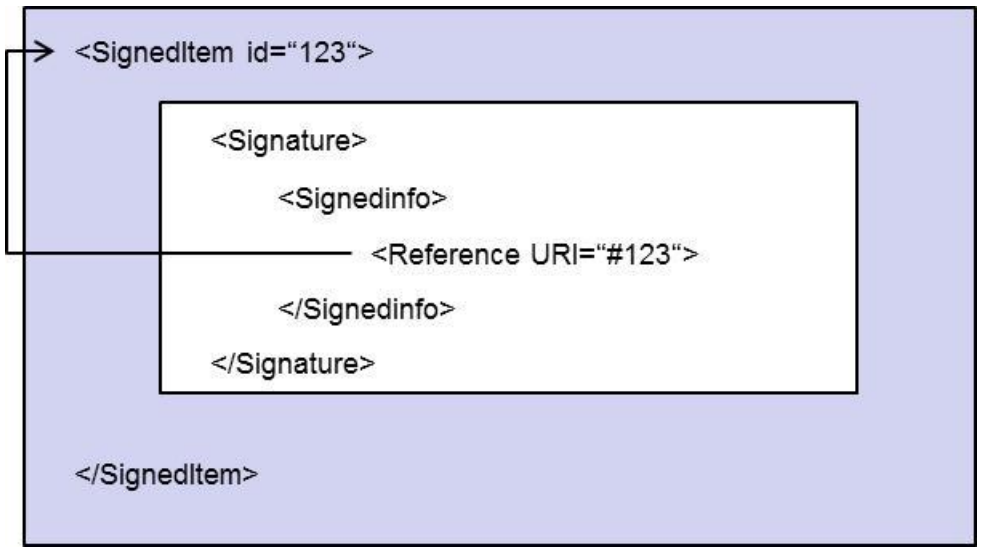


Abbildung 6 Enveloped Signature

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

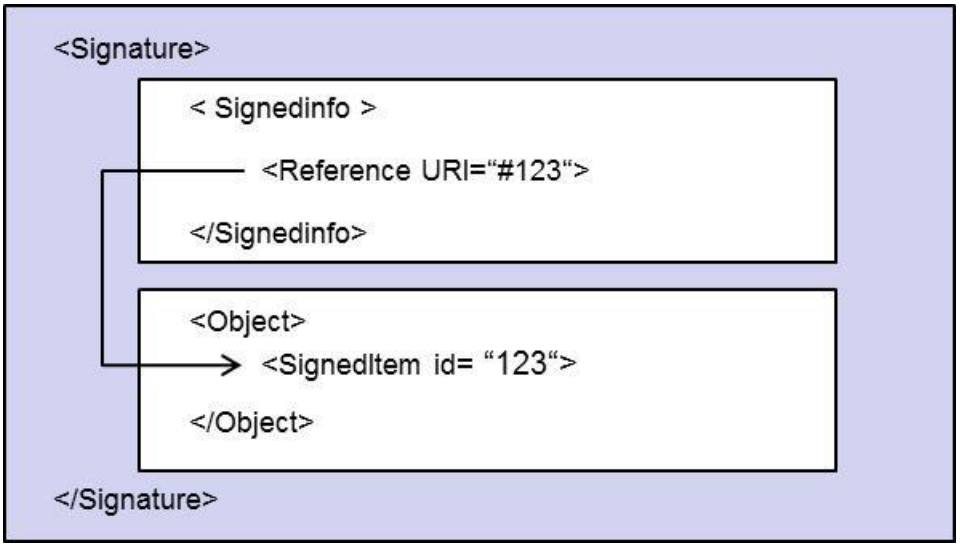


Abbildung 7 Enveloping Signature

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.

3.2 Risiken

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

3.2.1 Signatur

Beim Erstellen der Signatur besteht das Risiko, 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 folglich möglicherweise dessen Erscheinungsbild verändert werden, ohne dass dabei die Signatur unter dem vermeintlichen Dokument ungültig wird.

3.2.2 Verschlüsselung

Bei der Verschlüsselung besteht das Risiko, dass nicht alle zu schützenden 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 gezogen werden.

3.3 Signatur

Anmerkung zur folgenden Tabelle: Unter Anwendungsfälle werden Möglichkeiten aufgeführt, wie die im Kapitel 3.2.1erwähnten Risiken auftreten können.

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 4.1.3. «Behandlung der internen Verweise».

Nr.	Anwendungsfälle	Massnahme	Bemerkung
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-1 oder PDFA/A-2 Objekt umwandeln. Dieses Objekt muss dann vom Benutzer gemäss RFC 5652 signiert werden. Die beiden Dateiformate werden im Rahmen der öffentlichen Beurkundung von der EÖBV vorgeschrieben.	Zur Behandlung von Verweisen, s. auch Kapitel 4.1.3. «Behandlung der internen Verweise».

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:** Es soll kein ausführbarer Code in einem Verwaltungsdokument enthalten sein. Wenn Code im Dokument vorhanden ist, muss dieser von der Signatur ausgeschlossen werden. Ein Dokument mit solchem Inhalt soll vom Benutzer nicht signiert werden,	**Anmerkung:** Der hier beschriebene Fall soll 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 (Verwaltung)Dokument vorgenommen werden können, wobei die Signatur unter dem Dokument auch nach den Änderungen weiterhin gültig bleibt. S. auch Kapitel 4.1.2. «Enthaltene Programme im Dokument».

3.3.1 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 intern verwiesenen Objekte auch.	Bei den vollautomatisierten 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 4.2.7.3 «HMAC».

	Anwendungsfälle	Massnahme	Bemerkung
2	**Zusammenspiel zwischen XML-Anwendung und XML-Signaturprüfung.** Aufgrund der Flexibilität von XML besteht die Möglichkeit, die Signatur an einer anderen Stelle des Hauptobjekts einzubinden. Dabei kann/muss aber nicht die Signatur an Gültigkeit verlieren. Bei automatisch erstellten und geprüften XML-Signaturen, wie z.B. bei SAML oder SOAP, besteht dann die Gefahr, dass nicht die von der Signatur erfassten Bestandteile an die Applikation weitergegeben werden. Folglich akzeptiert die Applikation versehentlich Komponenten, welche durch die Signatur nicht geschützt sind. Eine Attacke wird als XML-Wrapping bezeichnet, wenn bewusst veranlasst wird, dass die Applikation - anders als vorgesehen - von der Signatur nicht erfasste Komponenten erhält, akzeptiert und weiterverarbeitet. Siehe hierzu z.B. folgende Papers [McAu], [Soetal]	**MUST:** Das XML-Schema ist zu prüfen. U.a., dass sich die XML Signatur wie auch die zu signierenden Objekte in dem vom Schema vorgesehen Ort befinden. Das zu verwendende Schema ist sicher, d.h. vor Veränderung oder Ersetzen, aufzubewahren. **SHOULD:** Das Schema, besser der Hashwert davon, soll Bestandteil der Signatur sein. **MUST:** Angenommen, die Anwendung hat sicherheitsrelevante und von der Signatur erfasste Informationen zu verarbeiten. Dann muss die Sicherheitsapplikation prüfen, ob diese Information von der Signatur erfasst wurde. Dies bevor sie diese Information an die Anwendung weiterreicht. **SHOULD:** Das, was innerhalb des Hauptobjekts zu signieren ist, soll mit einem absoluten XPATH-Pfad referenziert werden. Für die Referenz sollen nur Ausdrücke der Version 1.0 ohne Funktionen und Operatoren verwendet werden.	

	Anwendungsfälle	Massnahme	Bemerkung
3	**Denial of Service Attacke.** In XML Signature Best Practices Kapitel 2.1 sind eine Reihe von Denial of Services Attacken mit XML-Signaturen aufgelistet.	**SHOULD:** Die dort aufgeführten Massnahmen sollen befolgt werden, sofern entsprechende Schwachstellen vorliegen.	

3.4 XML Verschlüsselung

Anmerkung zur folgenden Tabelle: Unter Anwendungsfälle werden Möglichkeiten aufgeführt, wie die im Kapitel 4.2.2 erwähnten Risiken auftreten können.

3.4.1 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 z.B. über die TLS-Verbindung verschlüsselt übertragen, sondern über einen separate unverschlüsselte HTTP Verbindung ausgetauscht werden

3.4.2 Voll Automatisierte Verschlüsselung von XML-Dokumenten

Nr.	Anwendungsfälle	Massnahme	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 transportiert werden.	Beim automatisierten Datenaustausch ist es nicht erforderlich, dass alle Informationen zum Dokument wie z.B. die Schemas jedes Mal mitgereicht werden.

2	**Bei automatisch verschlüsselten Dokumenten** Ist das Schema eines verschlüsselten XML-Objekts bekannt, so lassen sich daraus Rückschlüsse auf den Klartext des verschlüsselten Objekts ziehen.	**SHOULD:** Das Schema zu einem verschlüsselten Objekts soll vertraulich gehandhabt werden.	

3.5 Signatur mit Verschlüsselung

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

Nr.	Anwendungsfälle	Massnahme	Bemerkung
1	**Austausch von verschlüsselten zuvor signierten XML-Dokumenten**. Der Einsatz von Detached Signature, welche nicht verschlüsselt werden, erlaubt u.a. eine beschleunigte Brute Force Attacke, weil eine beschleunigte Plausibilitätsprüfung für einen Kandidaten eines Entschlüsselungsschlüssels durchgeführt werden kann	**SHOULD:** Man soll auch die Detached Signature verschlüsseln, wenn das dazu gehörige unverschlüsselte Dokument oder Objekt Bestandteil der Signatur ist und chiffriert wird.	
2	**Austausch von (automatisch) verschlüsselten und zuvor signierten XML Objekten.** Falls von einer zu verschlüsselnden Komponente ein Hashwert besteht, und dieser nicht auch verschlüsselt wird, so erlaubt dies eine beschleunigte Plausibilitätsprüfung eines möglichen Entschlüsselungsschlüssels	**SHOULD:** Man soll die zu den verschlüsselnden Objekten entsprechenden Hashwerte ebenfalls verschlüsseln, sofern der Hashwert auf dem Klartext des zu verschlüsselnden Objekts basiert. Sind diese Hashwerte Bestandteil des zu verschlüsselnden Signaturobjekts, so sollen diese Werte ebenfalls verschlüsselt werden.	Siehe W3C XML Signature Best Practices

Nr.	Anwendungsfälle	Massnahme	Bemerkung
3	**Austausch von verschlüsselten und danach 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. (Eine solche Signatur stellt dann eine Beurkundung dar.) **SHOULD:** Das Dokument soll zuerst signiert, dann verschlüsselt und nachträglich wieder signiert werden. **SHOULD NOT:** Wenn ein verschlüsseltes Objekt signiert wird, dann soll die Signatur mit einem nach ZertES geregelten Zertifikat verifiziert werden können. **MUST:** Aus der Anwendung muss ersichtlich sein, dass diese Signatur nicht für den Inhalt (Content) der verschlüsselten Information einsteht, sondern nur für die Authentizität und Integrität beim Versand. (Analog zu einem Zeitstempel oder einer Empfangsbestätigung) **Anmerkung:** Das ZertES regelt die Herstellung der Zertifikate mit den entsprechenden Signaturen, die anderen Gesetze die Anwendung der Signatur.	Eine geregelte Signatur hat eine vom Gesetz erfasste rechtliche Implikation. Folglich besteht im Allgemeinen das Bedürfnis oder gar die Pflicht, das entsprechend Signierte aufzubewahren und später von berechtigten Dritten lesen zu können. Ist die Datei verschlüsselt, besteht diese Möglichkeit unter Umständen nicht mehr. Mit der Signatur nach der Verschlüsselung soll nur die Authentizität und Integrität geschützt werden, jedoch keine Beglaubigung vorliegenden Klartext vorgenommen werden. Folglich soll diese Signatur mit einem Zertifikat geprüft werden, welches kein Content Kommittent enthält.

4 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 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.

4.1 Vorverarbeitung des Dokuments

4.1.1 Separierung des Dokuments

Eine Verschachtelung von Signaturen innerhalb desselben Dokuments soll man möglichst vermeiden

SHOULD: Die zu signierenden Objekte sollen, falls möglich, separiert werden, d.h. in Teilobjekte zerlegt werden, damit nicht Verschachtelungen von Signaturen entstehen. Dabei soll 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.

4.1.2 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 sollen so gestaltet sein, dass sie im betreffenden Kontext oder Anwendung keine Programme (wie Makros, usw.) enthalten oder darauf referenzieren, welche bei der Präsentation des Verwaltungsdokuments ausgeführt werden.

Im entsprechenden Kontext sind die Dokumente auf Programminhalte in XML-Dokumenten zu prüfen. Die Prüfung soll 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, soll bei nicht vollautomatisierten Prozessen eine entsprechende Warnmeldung dem Benutzer angezeigt werden.

SHOULD NOT: Sind Programme im Dokument enthalten, soll 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.

4.1.3 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 hat, 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. Änderungen am Dokument haben aber eine ungültige Signatur zur Folge.

4.2 Signaturerstellung

Der Prozess der Signaturerstellung ist im Standard W3C-Sig beschrieben. Eine bildliche Darstellung dieses Prozesses ist in Anhang A dargestellt.

4.2.1 Signaturtypwahl

Bekanntlich stehen 3 Typen von XML-Signaturen zur Verfügung (Enveloped, Enveloping, Detached). Es gibt auch Mischformen zwischen Enveloping und Detached Signaturen.

MUST: Der Typ Enveloping oder Detached muss bei der Signatur über mehr als ein Objekt verwendet werden.

MUST NOT: Falls weitere von der Signatur erfasste Informationen,, wie sie bei ETSI EN 319 132-1 V1.1.1 aufgeführt sind, hinzugefügt werden, dann darf keine Enveloped Signatur verwendet werden.

4.2.2 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.

Über das Manifest Element lassen sich XML-Objekte, besser deren Hashwert, in die Signatur einbinden. Doch gemäss Standard kann die Signatur auch als gültig erachtet werden, wenn die im Manifest enthaltenen Hash-Werte nicht mehr mit den dort referenzierten Objekten übereinstimmen.

MUST NOT: Das Manifest Element darf nicht verwendet werden. Alles, was in die Signatur einfliesst, muss dann auch entsprechend gültig geprüft werden können.

4.2.3 Transformation der Objekte

Die Transformation wird im Standard W3C-Sig 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 «Canonicalization» der XML-Unterobjekte und des XML-Hauptobjekts muss durchgeführt werden. Für externe binäre Daten (engl. Binaries) wie Bilder ist eine Base64 Codierung anzuwenden.

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 im W3C-Sig Standard erwähnten «Canonicalization».
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 T transformiert wird und sich somit die daraus resultierende Prüfsumme nicht ändert. Deshalb soll der Benutzer sehen, was nach der Transformation signiert wird, siehe W3C-Sig Kapitel 8.1.3.

Weiterer Grund: Siehe 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 «Canonicalization» aufgeführt. Vereinfacht lässt sich sagen, dass die exklusive «Canonicalization» angewandt werden soll, wenn die Signatur aus dem Kontext herausgenommen werden und dabei ihre Gültigkeit erhalten bleiben soll. Kann eine Signatur aus ihrem Kontext entnommen werden und bleibt sie dann dennoch weiterhin gültig, so besteht die Gefahr einer XML-Wrapping-Attacke.

Konsequenz und wichtig: Zur Angabe der Elemente und deren Attribute **dürfen keine DTD Angaben** in den XML-Objekten gemacht werden, weil diese bei der «Canonicalization» 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.

4.2.4 Bemerkung zu kryptographischen Algorithmen

In den Standards von ENISA und des BSI sind Empfehlungen zu kryptographischen Verfahren aufgeführt, wie auch zu deren Schlüssellänge aufgeführt. Hier sind diejenigen Verfahren aufgeführt, welche sowohl bei den erwähnten Standards wie auch bei der W3C-Sig empfohlen werden. Sollen andere Verfahren als im Standard W3C-Sig eingesetzt werden, dann muss ein entsprechendes URI definiert und vereinbart werden.

4.2.5 Algorithmen für die Prüfsummen der Objekte

In den W3C Standards sind Algorithmen (Hashfunktionen) für den Message Digest (kryptographische Prüfsumme) aufgeführt. Hier wird wegen der Sicherheit präzisiert, welche anzuwenden sind.

Soll die Wahl des Verfahrens zur Herstellung der Prüfsumme (mit Hilfe eines standardisierten URI) angegeben werden, sind die Verfahren im entsprechenden W3C-Standard massgebend.

MUST: Nur die im W3C aufgeführten Hashfunktionen SHA-256, SHA-384, SHA-512 dürfen verwendet werden.

SHOULD: Für alle in die Signatur aufzunehmenden Dokumente soll der gleiche Algorithmus für die kryptographische Prüfsumme (engl. 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.

4.2.6 «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 «Canonicalization», welche im W3C-Sig Standard angegeben sind, sind zu unterstützen.

MUST NOT: Andere Verfahren zur «Canonicalization» dürfen aus Sicherheitsüberlegungen nicht angewandt werden.

Bevorzugt ist eine «Canonicalization», welche die Namensräume der zu signierenden Objekte berücksichtigt, d.h. sie beifügt, falls dort nicht vorhanden.

Je **im** Kapitel 8.1 des Standard [RFC 3275], und W3C-Sig, sowie im Kapitel 4.2.3 «Transformation der Objekte» sind Gefahren rund um die «Canonicalization» und Transformation beschrieben.

4.2.7 Verfahren für die Signatur

4.2.7.1 Algorithmen

Zur Wahl der Algorithmen für die Prüfsumme siehe Kapitel 4.2.5

SHOULD: Der Hash-Algorithmus für die Herstellung der Prüfsumme der Objekte soll mit dem Hash-Algorithmus für die Herstellung der Signatur identisch sein.

4.2.7.2 Asymmetrische Verfahren

MUSS: Im Benutzerumfeld muss beachtet werden, dass die Crypto Card die entsprechenden Algorithmen und Schlüssellängen zur Bildung der Signatur unterstützt. Anforderungen an die Crypto Card sind in TAV aufgeführt.

SHOULD: Im Server-Umfeld soll ein HSM-Modul für die sichere Aufbewahrung der privaten Schlüssel und für die Operationen mit diesen verwendet werden.

In den W3C Standards sind asymmetrische Algorithmen (Hashfunktionen) für den Message Digest (kryptographische Prüfsumme) aufgeführt. Hier wird wegen der Sicherheit präzisiert, welche anzuwenden sind.

MUST: RSA mit einer Schlüssellänge von mindestens 2048 Bit muss unterstützt werden.

SHOULD: DSA-Verfahren (Elliptische Kurven, diskreter Logarithmus) soll unterstützt werden.

MUST: Falls das DSA-Verfahren verwendet wird, muss beachtet werden, dass die Anzahl möglicher Werte im asymmetrischen Verfahren mindestens gleich gross ist wie die Anzahl möglicher Hashwerte.

Grund: Bereits bei der Bildung der Signatur können sonst Kollisionen entstehen. Die möglichen Hashwerte des Hashverfahrens werden auf die möglichen Werte des asymmetrischen Verfahrens reduziert. Im IETF RFC 6979 werden Beispiele erläutert, welche dem hier Beschriebenen widersprechen. Doch ein entsprechender Hinweis wie in diesem Dokument fehlt, dass solches nicht getan werden soll. Es handelt sich folglich um eine vom RFC 6979 unbeabsichtigte Auflistung von Falschbeispielen.

4.2.7.3 HMAC

Im W3C Standard sind die Algorithmen (Hashfunktionen) für die Authentizität mit HMAC aufgeführt. Die Sicherung der Authentizität ist gemäss Standard auch mittels HMAC erlaubt.

MUST: Nur die im W3C aufgeführten Hashfunktionen SHA-256, SHA-384, SHA-512 dürfen verwendet werden.

MUST NOT: Ist ein zu authentisierender Prozess nicht vollautomatisiert, dann darf das HMAC Verfahren nicht eingesetzt werden.

Grund: Das Schlüsselmanagement soll 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.

SHOULD NOT: Der Schlüssel für die Bildung des HMAC soll nicht der gleiche wie der für die Verschlüsselung sein.

MUST NOT: HMAC darf nicht für die Authentizität verwenden werden, wenn das entsprechende Objekt so aufbewahrt werden muss, dass die Gültigkeit der Authentizität nicht verlorengehen darf.

Grund: Der Schlüssel für die Erzeugung und Verifikation des HMAC-Werts müsste ebenfalls aufbewahrt werden. Es kann später nicht festgestellt werden, welche der Parteien den HMAC-Wert erzeugt hat.

4.2.8 Angaben zum Unterzeichnenden

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

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

Grund: Angaben in Zertifikaten sind (anerkannt) verlässlich.

Anmerkung: Andere Zertifikatsformen wie PGP oder SPKI Zertifikate werden in der Schweiz kaum verwendet und sind im Zusammenhang für die der Handunterschrift gleichgestellte elektronische Signatur und für die Erstellung eines elektronischen Siegels auch nicht erlaubt.

4.2.9 Anzeige an den Benutzer

Applikationen können dem Benutzer, welche die Signatur prüft, zusätzliche 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 soll bei unterschiedlichen Angaben nicht akzeptiert werden.

Grund: Siehe [RFC 3850] oder W3C-Sig Kapitel 8.1.

4.2.10 Angaben zu den Unterobjekten

Gemäss Standard W3C-Sig ist es erlaubt, Angaben zum Typ (MIME Type) der zu signierenden Objekte beizufügen.

SHOULD: Angaben zum Typ der zu signierenden Objekte sollen 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.

4.2.11 Zeitangabe

Nicht anerkannte Zeitangaben zur Signatur, z.B. vom Signierenden erfolgte Zeitangaben, sind wenig sinnvoll, denn eine Zeitangabe in diesem Kontext will etwas belegen oder gar beweisen. Anerkannte Zeitangaben werden mittels eines Zeitstempels einer unabhängigen anerkannten Stelle erstellt.

MUST: Es dürfen nur nach ZertES qualifizierte Zeitstempel verwendet werden, welche von einer nach ZertES anerkannten CSP (Zertifizierungsdienstanbieter) ausgestellt werden (Art. 2 Bst. j ZertES).

Grund: Nur diese Zeitstempel sind gewiss anerkannt.

MUST: Falls belegt werden soll, dass eine Signatur vor oder nach einem bestimmten Zeitpunkt erstellt worden ist, dann müssen die entsprechenden XML-Elemente im ETSI Standard EN 319 132-1 V1.1.1 verwendet und in die dort vorgegebene Struktur (Schema) eingebettet werden (AllDataObjectsTimeStamp, SignatureTimeStamp).

Gegensignatur

SHOULD: Eine Gegensignatur soll - wie im ETSI-Standard EN 319 132-1 V1.1.1 beschrieben - erstellt und an entsprechender Stelle eingefügt werden.

4.2.12 Weitere Angaben

SHOULD: Werden zusätzliche Informationen, wie sie in ET319 132-1 V1.1.1 zur Signatur oder zum Signierenden aufgeführt und sind diese Informationen Bestandteil der Signatur, so soll dies nach dem soeben genannten Standard erfolgen.

4.3 Verschlüsselung

4.3.1 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 (vertraulichen) 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, sollen ebenfalls verschlüsselt werden.

Grund: Die Prüfsummenwerte oder Signaturen sollen 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.

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

4.3.2 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.

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

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

4.3.3 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 soll, 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 gemäss dem Standard CMS (Cryptographic Message Syntax [RFC 5652]) verschlüsseln.
- 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 sollen nie verändert werden. Müssen aber die signierten Objekte aus irgendwelchen Gründen vor der Verschlüsselung doch transformiert werden, dann muss gelten:

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

Grund: Im Unterschied zur Signatur verändert die Aufbereitung (die «Canonicalization») 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 sollen, gegebenenfalls zuerst für die Verschlüsselung aufbereiten, dann signieren und dann erst verschlüsseln. Zur Problematik der «Canonicalization» im Zusammenhang mit der Signatur sind weitere Informationen bei http://www.w3.org/Security/ aufgeführt.

Beispiel für Inkompatibilität und Standard konformer Anwendung: Bei der «Canonicalization» 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 der «Canonicalization» angewandt, welche die Kommentare entfernt. Konsequenz ist, dass die Signatur nach der Entschlüsselung nicht mehr gültig ist.

4.3.4 Algorithmen für die Verschlüsselung

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

SHOULD NOT: Wird nach der Verschlüsselung das Objekt noch mit einer Authentisierungsinformation (HMAC, Signatur) versehen, soll der Galois Counter Mode (GCM) nicht verwendet werden.

Grund: Zur Authentisierung und Verschlüsselung sollen grundsätzlich andere Schlüssel verwendet werden, sieh auch Kapitel 4.4 XML Signature Best Practices.

4.3.5 Schlüsselvereinbarung/-einigung (Key Agreement)

Es besteht gemäss W3C-Standard zur XML-Verschlüsselung auch noch die Möglichkeit, mit der XML-Signatur- und XML-Verschlüsselungs-Struktur Schlüssel auszutauschen oder sich auf einen Schlüssel zu einigen.

SHOULD NOT: Key Agreement (Schlüsselvereinbarung)

Grund: Hierzu bestehen andere, benutzerfreundliche und vor allem standardisierte Technologien wie

TLS, als XML-Objekte dafür einzusetzen.

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

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

4.3.6 Schlüsseltransport

SHOULD NOT: Zum Schlüsseltransport soll RSA PKCS1 Version 1.5 - anders als im W3C Standard aufgeführt - nicht unterstützt werden, siehe dazu BSI TR-02102-1, Seite 19.

SHOULD: RSA-OAEP soll unterstützt werden, siehe dazu BSI TR-02102-1, Seite 19.

Zu RSA-OAEP und RSA PKCS 1 siehe RFC 3447.

5 Alternativen

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. Dies wiederum kann zu erheblichen Schwierigkeiten in den praktischen Anwendungen und Umsetzungen führen.

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, die Daten bei Bedarf anzeigen und sicherheitsrelevante Veränderungen feststellen kann.

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 5652].

Der Nachteil bei der Signatur eines ZIP-Files im Benutzerumfeld ist, dass sich der Benutzer nicht genau anzeigen lassen kann, was er signiert hat, nämlich lediglich das ZIP-File. Dem Benutzer wird nämlich nicht das ZIP-File angezeigt, sondern dessen dekomprimierter 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.

6 Sicherheitsüberlegungen

Keine weiteren

7 Haftungsausschluss/Hinweise auf Rechte Dritter

eCH-Standards, welche der Verein **eCH** dem Benutzer zur unentgeltlichen Nutzung zur Verfügung stellen oder welche **eCH** referenzieren, 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.

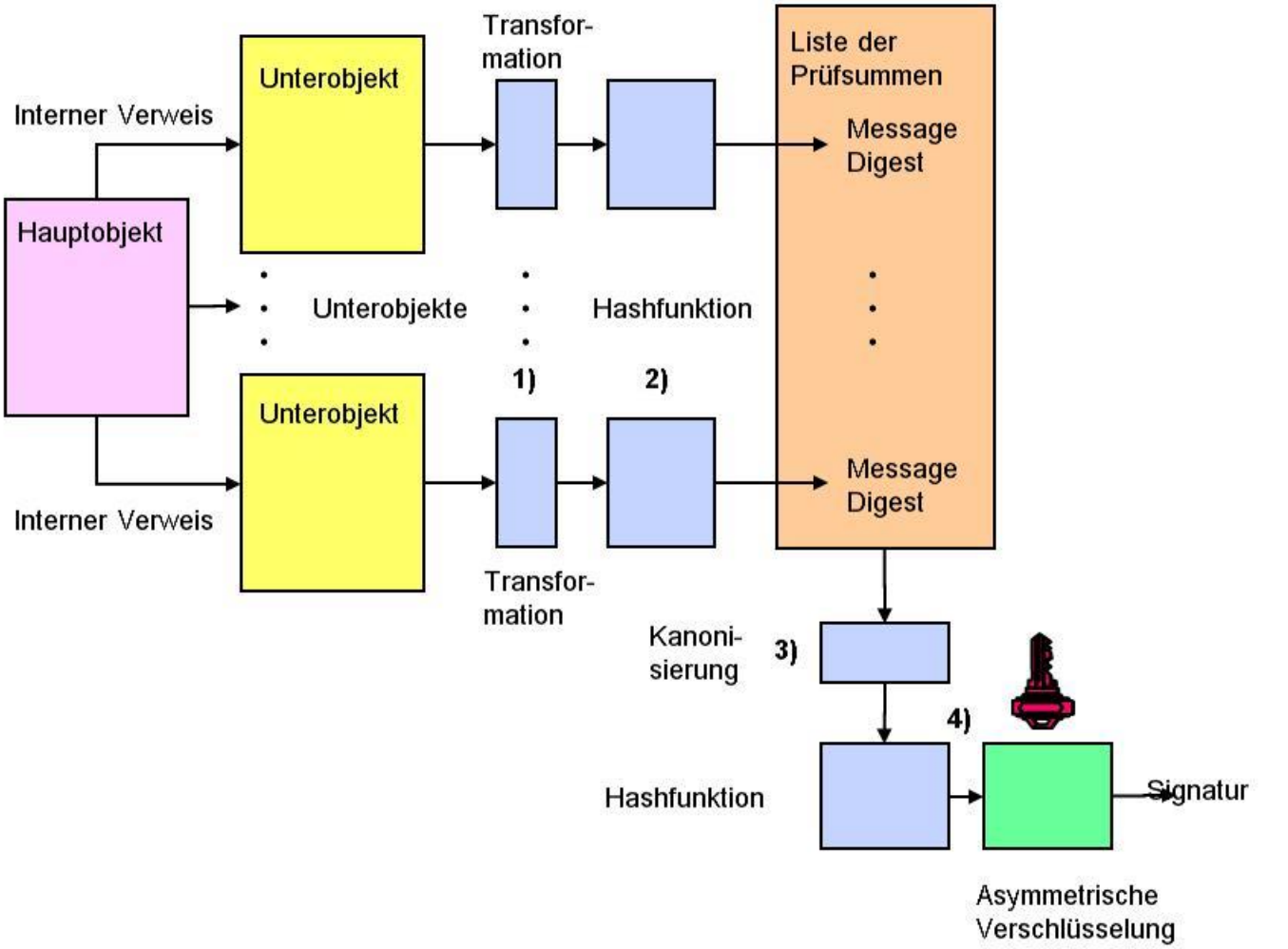
Die von den Fachgruppen erarbeiteten Standards können unter Nennung der jeweiligen Urheber von **eCH** unentgeltlich und uneingeschränkt genutzt, weiterverbreitet und weiterentwickelt werden.

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:



a

Abbildung 8 Signaturbildung

Die Herstellung der Signatur erfolgt in 4 Schritten:

1. Sämtliche der zu signierenden Objekte werden einer oder mehrerer Transformationen unterzogen, wobei gemäss W3C-Sig Standard 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 W3C-Sig Standard 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 «Canonicalization» wird dann die Signatur hergestellt. Gemäss W3C-Sig Standard 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 wird.
- Angaben zur Entität, welche die Signatur hergestellt hat.

Anhang B – Referenzen & Bibliographie

Fachliteratur

[McAu]	Michael MacIntosh, Paula Austel, XML Signature Wrapping Attacks and Countermeasures
[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/
[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
[Soetal]	Juray Somorovsky, Andreas Mayer at al, On Breaking SAML: Be Whoever You Want to be

eCH (www.ech.ch)

eCH-0018	XML Best Practices
eCH-0036	Dokumentation für den XML-orientierten Datenaustausch

ETSI (www.etsi.org))

ETSI TS 119 102-1 V1.2.1	Electronic Signatures and Infrastructures (ESI); Procedures for Creation and Validation (2018-08)
ETSI EN 319 132-1 V1.1.1	Electronic Signatures and Infrastructures (ESI); XAdES digital signatures; Part 1: Building blocks and XAdES baseline signatures

IETF Standards (www.ietf.org)

RFC 3023	XML Media Types
RFC 3076	Canonical XML Version 1.0
RFC 3161	Internet X.509 Public Key Infrastructure Time Stamp Protocol (TSP)
RFC 3275	XML Signature Syntax and Processing
RFC 3447	Public Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1
RFC 3741	Exclusive XML Canonicalization, Version 1.0
RFC 3986	Uniform Resource Identifier (URI): Generic Syntax
RFC 4452	The "info" URI Scheme for Information Assets with Identifiers in Public Namespaces
RFC 5652	Cryptographic Message Syntax (CMS)
RFC 6979	Deterministic Usage of the Digital Signature Algorithm (DSA) and Elliptic Curve Digital Signature Algorithm (ECDSA)

W3C Standards (www.w3C.org)

Canonical XML Version 1.0 und 1.1 Recommendation March 2001 and May 2008

Decryption Transforms for XML Signature Recommendation, December 2002

Describing Media Content of Binary Data in XML W3C Working Group Note, May 2005

Exclusive XML Canonicalization Version 1.0 Recommendation, July 2002

XML Encryption and Syntax Processing Recommendation, Version 1.1, April 2013

XML Path Language (XPath) Version 1.0

XML Schema Part 1: Structures Second Edition. 28 October 2004

XML Schema Part 2: Datatypes Second Edition. 28 October 2004

XML Signature Best Practices Working Group Note, April 2013

XML Signature Syntax and Processing Recommendation Version 1.1, 11 April 2013

XSLT 2.0 and XQuery 1.0 Serialization (Second Edition) Recommendation, December 2010

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

ENISA (www.enisa.europa.eu)

Algorithms, key size and parameters report - 2014, November 2014, European Union Agency for Network and Information Security Agency

BSI (www.bsi.de)

Kryptographische Verfahren: Empfehlungen und Schlüssellängen, BSI TR-02102-1, Version: 2019-01, Stand: 22. Februar 2019

OASIS Standards (www.oasis-open.org)

Security Assertion Markup Language (SAML) v2.0

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

Web Services Security, SOAP Messages Security 1.1, February 2006

Erlasse

EÖBV: Verordnung des EJPD über die Erstellung elektronischer öffentlicher Urkunden und elektronischer Beglaubigungen vom 8. Dezember 2017, SR 211.435.11

TAV: Technische und administrative Vorschriften über Zertifizierungsdienste im Bereich der elektronischen Signatur und anderer Anwendungen digitaler Zertifikate vom 23. November 2016, SR 943.032.1

UIDG: Bundesgesetz über die Unternehmens-Identifikationsnummer vom 18. Juni 2010, SR 431.03

VZertES Verordnung über Zertifizierungsdienste im Bereich der elektronischen Signatur und anderer Anwendungen digitaler Zertifikate vom 23. November 2016, SR 943.032

ZertES: Bundesgesetz vom 18. März 2016 Bundesgesetz über Zertifizierungsdienste im Bereich der elektronischen Signatur und anderer Anwendungen digitaler Zertifikate, SR 943.03

Anhang C – Mitarbeit & Überprüfung

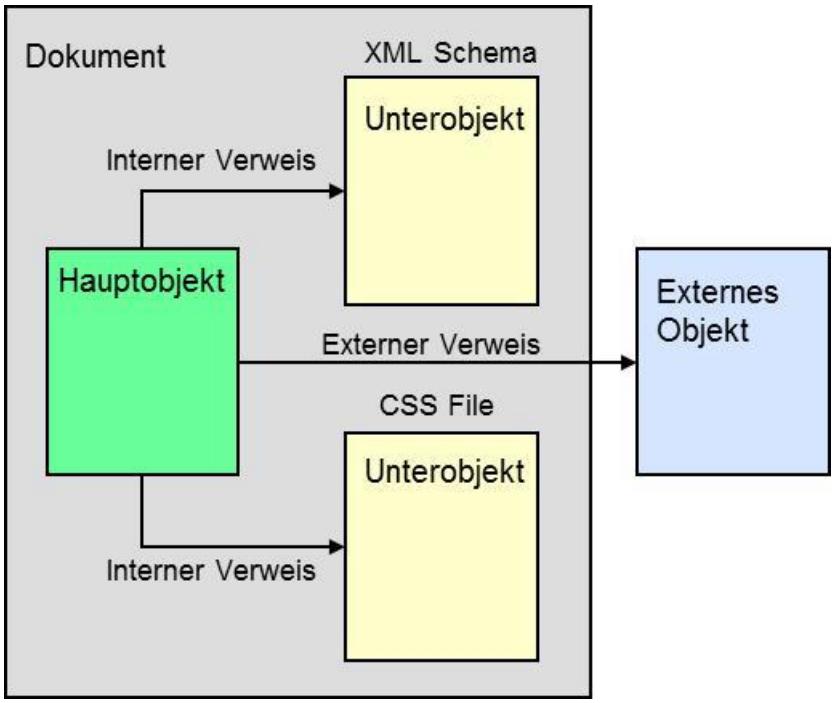
Siehe Titelblatt

Anhang D – Abkürzungen und Glossar

Glossar

«Canonicalization»	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 «Canonicalization» auf das XML-Objekt an. Nach der Kanonisierung setzt sich dieses Objekt auf den verschiedenen Systemen Byte für Byte gleich zusammen. Standards zur «Canonicalization» sind z.B. [RFC 3076], [RFC 3741] oder die W3C-Standards dazu.
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.
Geregeltes elektronisches Siegel	Gemäss Art. 2 Bst. d ZertES, eine fortgeschrittene elektronische Signatur, die unter Verwendung einer sicheren Siegelerstellungseinheit nach Artikel 6 ZertES erstellt wurde und auf einem geregelten, auf eine UID-Einheit nach Artikel 3 Absatz 1 Buchstabe c des Bundesgesetzes vom 18. Juni 2010 über die Unternehmens-Identifikationsnummer (UIDG) ausgestellten und zum Zeitpunkt der Erzeugung des elektronischen Siegels gültigen Zertifikat beruht;
Hauptobjekt	Ursprungsobjekt, von welcher die expliziten Verweise starten. Objekt, welches das Wurzelelement enthält.

Qualifizierte Signatur	Siehe Art. 2 Bst. e ZertES: Eine geregelte elektronische Signatur, die auf einem qualifizierten Zertifikat beruht.
Qualifiziertes Zertifikat	Ein digitales Zertifikat, das die Anforderungen des Artikels 8 ZertES erfüllt.
Signaturtypen	Es sind folgende Signaturtypen standardisiert und in der Technik am weitesten verbreitet: a. Cryptographic Message Syntax RFC 5652, Ältere Versionen davon sind: PKCS#7 Signature s. [RFC 2315], neuer Standard bei IETF CMS (Cryptographic Message Syntax [RFC 3852] b. XML Signature s. W3C-Sig 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 W3C-Sig 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 impli-zierter Verweis referenziert auf ein nicht existierendes Objekt oder eine nicht ausgeschriebene Steueranweisung, sondern auf etwas, was inner-halb 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 Ele-ment innerhalb des Elements «Webseite», welches mit <html: > beginnt wird als Information in HTML interpretiert. Bei den expliziten Verweisen wird zwischen Dokument internen und ex-ternen Verweisen unterschieden. Dokument interne Verweise sind Ver-weise auf Unterobjekte (z.B. Dateien), welche Bestandteil des Doku-ments sind, wie Schemas oder Bilder. Externe Verweise sind jedoch Ver-weise auf externe Objekte (Haupt- oder Unterobjekte), welche Zusatzin-formationen enthalten, aber für die Interpretation des Dokuments nicht wesentlich sind. Beispiel eines externen Verweises ist eine Quellenan-gabe, welche z.B. in HTML oder XML geschrieben ist. Hierzu folgende Il-lustration:  **Anmerkung:** Was nun als einen externen oder internen Verweis betrach-tet wird, hängt einerseits von den Intentionen des Verfassers des XML-Dokuments, von der Kommunikationsgemeinschaft oder von der Kommu-nikationsplattform ab.
XML-Dokument	Ein Dokument, dessen Hauptobjekt eine XML-Struktur aufweist.
XML-Verwal-tungsdokument	Ein Verwaltungsdokument, dessen Hauptobjekt eine XML-Struktur auf-weist.

Abkürzungen

Abs.	Absatz
Bst.	Buchstabe
CSS	Cascading Style Sheets Language
DTD	Document Type Definition
ETSI	European Telecommunications Standards Institute
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
SAML	Security Assertion Markup Language (SAML) v2.0
SOAP	Service Oriented Application Protocol
SOAP	Simple Object Access Protocol
SSL	Secure Socket Layer
TAV	Technische und administrative Vorschriften über Zertifizierungsdienste im Bereich der elektronischen Signatur und anderer Anwendungen digitaler Zertifikate vom 23. November 2016, SR 943.032.1
TLS	Transport Layer Security
URI	Universal Resource Identifier
W3C	World Wide Web Consortium
W3C-Sig	XML Signature Syntax and Processing Recommendation Version 1.1, 11 April 2013
XHTML	Extensible Hypertext Markup Language
XLink	Extensible Linking Language
XML	Extensible Markup Language
XSLT	XSL Transformations

Anhang E – Änderungen gegenüber Vorversion

Request	Kapitel	Seite	Anpassung
	Titelblatt		Von Best Practice zum Standard. Grund dafür, siehe Anmerkung unten.
	2.1	7	Auflistung geändert (Anhang B weggelassen)
	3.3.1	16	Hinzufügen der Zeile 2 und 3 in der Tabelle
	3.4.2	19	Hinzufügen der Zeile 2 in der Tabelle
	3.5	20	Hinzufügen der Zeile 3 in der Tabelle
	4.2.1	23	«MUST NOT» Empfehlung zugefügt
	4.2.3	24	Absatz «Grund» gestrichen zur Transformation der Objekte
	4.2.4	25	Hinzugefügt
	4.2.7.3	26	Ergänzung zu den asymmetrischen Verfahren und zur Aufbewahrung der privaten Schlüssel
	4.2.11	27	Ergänzende Empfehlung
	4.2.12	28	Ergänzende Empfehlung
	4.3.4	29	Ergänzende Empfehlung
	4.3.5	29	«Schlüsselvereinbarung …» nach hinten geschoben
	4.3.6	30	Ergänzende Empfehlung
	Anhang B		Ergänzung/Streichung bei der Fachliteratur, Hinzufügen von ETSI und IETF RFCs, Aktualisierung der W3C Standards, Hinzufügen eines BSI und ENISA-Standards, Erweiterung der Angaben zu den Erlassen
	Anhang D		Ergänzung im Glossar
	Anhang D		Anpassung des Abkürzungsverzeichnis

Anmerkung: Das Dokument wurde von Best Practice zum Standard aufgewertet. U.a. aufgrund der Bedeutung des hier behandelten Themas, soll es ein Standard werden. Eine der hier abgehandelten Probleme, welche sich aufgrund der Verlinkungen von XML-Objekten ergibt, wird als eines der Top-10 Themen bei der OWASP aufgeführt. Siehe dort Item 3 und 4 https://owasp.org/www-project-top-ten/.

Weiter wird die Problematik der Präsentation eines XML-Dokuments an den Anwender erläutert und Lösungsvorschläge aufgezeigt. Dies wird im entsprechenden XML-Standard von W3C nicht abgehandelt. In diesem Zusammenhang lassen sich in Analogie dazu Empfehlungen zu E-Mail Signaturen mit HTML-Inhalt ableiten.

In diesem Dokument ist - anders als zu den erwähnten Standards - u.a. zusätzlich aufgeführt, dass bei der Verschlüsselung nach der Signatur Inkompatibilitäten zwischen den W3C-Standards bestehen. Darauf weisen aber diese Standards nicht hin.

Zudem sind Empfehlungen zur Zeitangabe und zur Prüfung einer Signatur gemacht worden, Die Empfehlungen zur Prüfung der Signatur lassen sich nicht in einem Standard finden, sind aber in wissenschaftlichen Publikationen zu finden, siehe z.B. [Soetal].

Anhang F – Abbildungsverzeichnis