ETSI TR 102 038 V1.1.1 (2002-04)

Technical Report

TC Security - Electronic Signatures and Infrastructures (ESI); XML format for signature policies

Reference

DTR/SEC-004011

Keywords

electronic signature, security

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

Individual copies of the present document can be downloaded from:
http://www.etsi.org

The present document may be made available in more than one electronic version or in print. In any case of existing or perceived difference in contents between such versions, the reference version is the Portable Document Format (PDF). In case of dispute, the reference shall be the printing on ETSI printers of the PDF version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at
http://portal.etsi.org/tb/status/status.asp

If you find errors in the present document, send your comment to:
editor@etsi.fr

Copyright Notification

Contents

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://webapp.etsi.org/IPR/home.asp).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Technical Report (TR) has been produced by ETSI Technical Committee Security (SEC).

Introduction

Electronic commerce is emerging as the future way of doing business between companies across local, wide area and global networks. Trust in this way of doing business is essential for the success and continued development of electronic commerce. It is therefore important that companies using this electronic means of doing business have suitable security controls and mechanisms in place to protect their transactions and to ensure trust and confidence with their business partners. In this respect the electronic signature is an important security component that can be used to protect information and provide trust in electronic business.

The European Directive on a community framework for Electronic Signatures defines an electronic signature as: "data in electronic form which is attached to or logically associated with other electronic data and which serves as a method of authentication". An electronic signature as defined in TS 101 733 [1] is a form of advanced electronic signature as defined in the Directive.

TS 101 733 [1] defines formats for electronic signatures that are compliant with the European Directive. Currently, the ETSI standard uses Abstract Syntax Notation 1 [ASN.1] to define the structure of the electronic signature. The structure of the electronic signature defined in TS 101 733 [1] is based on the structure defined in RFC 2630 [2]:"Cryptographic Message Syntax" (RFC 2630 [2]). TS 101 733 [1] satisfy the requirements made by the European Directive by defining new ASN.1 structures that can be added as parts of the fields *"signedAttrs"* and *"unsignedAttrs"*.

As a consequence of the growing importance of the use of XML on Internet, a standard for XML based digital signatures is currently being produced within W3C and IETF Working Group "XML-Signature Core Syntax and Processing" [8]. ETSI is in the process of producing a technical specification [5] that defines a XML format for electronic signatures that are compliant with the European Directive, as TS 101 733 [1] does for ASN.1 syntax. An electronic signature produced in accordance with that document provides evidence that can be processed to get confidence that some commitment has been explicitly endorsed under a Signature policy, at a given time, by a signer under an identifier, e.g. a name or a pseudonym, and optionally a role.

TS 101 733 [1] also deals with the Signature Policy issue. Although the present document does not mandate any form of Signature Policy specification, it specifies an ASN.1 based syntax that may be used to define a structured Signature Policy in a way that machines can read and process.

The present report deals with the specification of new XML elements able to contain the Signature Policy information specified in TS 101 733 [1].

1 Scope

The present document represents a very first version of a XML format for Signature Policies able to contain information on Signature Policies as specified by TS 101 733 [1]. The specifications given being so preliminary, a number of open issues for discussion and even definitions appear throughout the document.

Successive versions will gradually improve the new XML types defined aligning them with current efforts in the XML arena (specially those in the RDF [6] and [7], P3P [8] and [9] and Certification Practice Statements fields).

The present document:

Contains a mention to current efforts in the XML arena that are strongly related with the object of the present document, namely RDF, P3P and XML formats for CSP.

Contains a short description of the approach taken for the production of the present document and the approach to be adopted in order to align its contents to the results of the aforementioned work in XML arena. This would likely imply that the final specification will be somehow different to the initial one.

Contains a first specification for a XML format of a Signature Policy, mainly based on the contents of the ASN.1 structures defined in TS 101 733 [1]. However, a number of open issues to discuss and even explicit usage of XML types defined elsewhere (mainly in RDF and P3P), appear, signalling how the specifications will evolve to achieve a more tight alignment with current practices in XML world.

The rest of the document is structured as follows:

Clause 2 shows the relevant references for the present document.

Clause 4 mentions relevant related work being done at the time the present document has been produced, namely RDF and P3P.

Clause 5 outlines the technical approach followed to produce the present document.

Clause 6 presents the concepts of signature policy and signature validation policy.

Clause 7 shows the namespace definitions for following XML schema definitions.

Clause 8 shows the details of the XML schema definitions for the elements able to contain computer processable information of the signature policy. It also contains the rationale for each of the specified elements.

Finally, clause 9 introduced a set of initial comments on ways of improvement of the specifications given in the present technical report.

2 References

For the purposes of this Technical Report (TR) the following references apply:

[1] ETSI TS 101 733: "Electronic Signature Formats".

[2] RFC 2630 (June 1999): "Cryptographic Message Syntax".

[3] RFC 2459: "Internet X.509 Public Key Infrastructure Certificate and CRL Profile".

[4] W3C 08-2001 (W3C/IETF Proposed Recommendation, August 2001): "XML-Signature Syntax and Processing".

[5] ETSI TS 101 903: "XML Advanced Electronic Signatures (XAdES)".

[6] W3C 2-1999 (W3C Recommendation, 22 February 1999): "Resource Description Framework (RDF) Model and Syntax Specification".

NOTE: URL: http://www.w3.org/TR/REC-rdf-syntax.

[7] W3C 3-2000 (W3C Candidate Recommendation, 27 March 2000): "Resource Description Framework (RDF) Schema Specification 1.0".

NOTE: URL: http://www.w3.org/TR/2000/CR-rdf-schema-20000327.

[8] W3C 10-2000 (W3C Working Draft, 18 October 2000): "The Platform for Privacy Preferences 1.0 (P3P1.0) Specification".

NOTE: URL: http://www.w3.org/TR/P3P/.

[9] W3C 02-2001 (W3C Working Draft, 26 February 2001): "A P3P Preference Exchange Language 1.0 (APPEL 1.0)".

NOTE: URL: http://www.w3.org/TR/P3P-PREFERENCES".

[10] RFC 2459 (1998): "Proposed TLA and NLA Assignment Rule".

[11] RFC 2560 (1999): "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP".

3 Abbreviations

APPEL 1.0	A P3P Preference Exchange Language 1.0e
ASN.1	Abstract Syntax Notation 1
CA	Certificate Authority
CAD	Card Accepting Device
CRL	Certificate Revocation List
P3P	Platform for Privacy Practices Project
RDF	Resource Description Framework
XAdES-T	XAdES with Timestamp
XML	eXtensible Markup Language

4 Relevant related work in the XML arena

The following clauses mention some of the works in course within the XML arena that can have an impact on the Signature Policy XML format development, in the view of the author. They constitute only a note of attention devoted to launch a debate on the extent of the aforementioned impact that should eventually lead to define an agreed technical approach.

4.1 RDF and the Semantic Web

RDF "is a foundation for processing metadata; it provides interoperability between applications that exchange machine-understandable information on the Web" [6]. It can be used then "in resource discovery (…) by intelligent software agents" [6] and others contexts.

RDF [6] defines a model able to assign named properties and property values to resources in the Web by means of the production of statements. RDF model could be used by any syntax. However, RDF recommendation specifies a XML syntax for serialization and exchange of the models: RDF syntax. RDF [7] also specifies the mechanisms needed to define the relevant metadata to any domain, "to define the classes of resources they may be used with, to restrict possible combinations of classes and relationships, and to detect violations of those restrictions". It provides, in summary, "a type system for use in RDF models".

Signature Policies are pieces of data issued by certain organizations with authority to do it. RDF framework can be used, at least, to define a way of issuing information on the different Signature Policies defined all over the world, so that all of them will be available to any community desiring to know the particularities of each one.

When dealing with the impact of RDF on Signature Policies, one could think in defining XML RDF schemas to provide information on certain properties of the Signature Policy (information related with who issues it, validity period, location of the document where the Policy is defined, etc…). However, under a broader perspective, one could even think in define RDF schemas for certain parts of the Signature Policy contents themselves. A careful study should be made in order to assess the impact of RDF on the specification of XML Format for Signature Policies.

4.2 P3P and the explicitation of preferences and rules

The Platform for Privacy Practices Project (P3P) "enables Web sites to express their privacy practices in a standard format that can be retrieved automatically and interpreted easily" [8]. P3P defines "the syntax and the semantics of P3P privacy policies", which consist of "statements made using the P3P vocabulary for expressing privacy practices" [8]. P3P provides XML definitions for containing precise description of legal entities "making the representation of the privacy practices" [8]. It also provides with mechanisms to make statements on the privacy practices, including structured or unstructured data. It also defines a mechanism (Data Schema) that allows to any community to describe specific data elements and hierarchically group them.

A careful study should be made in order to assess the impact of the different mechanisms and XML types and elements defined by P3P on the specification of XML Format for Signature Policies. At a very first level, one can concludes that there is no use in re-defining all the information set dealing with the identification of legal entities, dates, etc. But the study should go beyond on that, and take into consideration the possible use of those mechanisms specified in P3P to extend the set of data elements. As an example, and apart from the immediate possibility of taking the ENTITY element defined there, P3P has specified elements to include information on dispute resolution procedures, which, as it has been pointed out by some comments, is currently missing in the TS 101 733 [1] specification, and in the present document.

Having said that, comments have also been raised on the fact that some of the elements in P3P use the XML attributes as a way of structuring information instead of XML elements, as shown below:

```
<ENTITY>
  <DATA-GROUP>
    <DATA ref="#business.contact-info.postal.city">CityEx"</DATA>
    <DATA ref="#business.contact-info.postal.stateprov">"ProvEx"</DATA>
  </DATA-GROUP>
</ENTITY>
```

In the former example the XML attribute `ref` also contains information that somehow can be considered part of a structure.

As a companion of P3P, the Web Consortium is working on the specification of the APPEL 1.0 (A P3P Preference Exchange Language 1.0e) language "for describing collections of preferences regarding P3P policies between P3P agents" [9]. This language would allow a user to "express her preferences in a set of preferences-rules () which can then be used by her user agent to make automated or semi-automated decisions" [9].

A study should be carried out in order to assess the impact of the existence of such a language on the Signature Policies XML format.

4.3 RDF and P3P relationship

P3P is not defined using RDF syntax, however, the group admits that privacy policies "may also be represented using the RDF data model" and says that "an RDF representation is not included in the present document. (Such a representation is planned to be made available as a W3C Note prior to submitting P3P as a Proposed Recommendation, together with a suitable RDF encoding of the policy reference file)". In summary, no strict alignment in syntactic terms exists between both documents, although the RDF model is not, strictly speaking, restricted to any syntax, and in consequence, a RDF model can be built for P3P.

The archives of the P3P group emails contain some work done on this subject, with a definition of a RDF model for P3P and its codification in XML. Obviously, in relation with the impact of the present report, more work on this particular issue has to be made.

5 Technical Approach

The present document firstly aims to define a XML Format for specifying Signature Policies able to contain the information specified within TS 101 733 [1]. However, in a broader sense, the XML Signature Policies format should also take into account the XML community to which is devoted. XML is strongly related with working in Web environments, and in consequence, it should accommodate to new requirements and/or ways of doing things. This implies that a definition of a XML type for containing equivalent pieces of information as the ones defined in TS 101 733 [1] could not be enough to take benefit of all the powerfulness of the different XML standards, constraining in this way the usage of such a specification itself.

Taking this into consideration the author thinks that the technical approach to the specification of the XML format for Signature Policies should be the following one:

As a first step, the exercise of taking the data structures in TS 101 733 [1] and specify new XML schema definitions for XML elements able to contain such pieces of information should be done. **This stage was initially covered in the version v0.0.1 (April 2001)**.

Secondly, studies on the RDF and P3P capabilities and potential impacts on the XML format for Signature Policies should be carried out. Such studies should focus on potential impacts at different levels: one thing is to use certain structures in RDF to give metadata on a XML document containing the Signature Policy and other thing is to model the contents of the Signature Policy itself in RDF; one thing is to use data structures defined in P3P to identify legal entities, for instance, and other thing is to use the data schema definition mechanism provided by P3P to make use of it within the Signature Policy document.

The aforementioned studies would launch, in this way, work on the alignment of the XML format for Signature Policies with the premises of the Semantic Web, and would eventually lead to a much more well adapted document to the XML environment.

6 Signature Policy and Signature Validation Policy

The **Signature Policy** is a set of rules for the creation and validation of an electronic signature, under which the signature can be determined to be valid. A given legal/contractual context may recognize a particular signature policy as meeting its requirements.

The signature policy may be explicitly identified or may be implied by the semantics of the data being signed and other external data like a contract being referenced which itself refers to a signature policy.

An explicit signature policy has a globally unique reference, which is bound to an electronic signature by the signer as part of the signature calculation.

The signature policy needs to be available in human readable form so that it can be assessed to meet the requirements of the legal and contractual context in which it is being applied. To facilitate the automatic processing of an electronic signature the parts of the signature policy which specify the electronic rules for the creation and validation of the electronic signature also needs to be in a computer processable form.

The signature policy thus includes the following:

rules, which apply to functionality, covered by the present document (referred to as the **Signature Validation Policy**);

rules which may be implied through adoption of Certificate Policies that apply to the electronic signature (e.g. rules for ensuring the secrecy of the private signing key);

rules, which relate to the environment used by the signer, e.g. the use of an agreed CAD (Card Accepting Device) used in conjunction with a smart card.

The Signature Validation Policy includes rules regarding use of Trusted Service Providers (CA, Attribute Authorities, Time Stamping Authorities) as well as rules defining the components of the electronic signature that shall be provided by the signer with data required by the verifier to provide long-term proof.

The rules to be followed by the signer appear in the `SignerRules` element (see clause 8.7.1).

The rules to be followed by the verifier appear in the `VerifierRules` element (see clause 8.7.2).

The rules for the use of CAs appears in the `SigningCertTrustCondition` element (see clause 8.8). Firstly, these rules include rules for certpath management, which appear in the `SignerTrustTrees` element (see clause 8.8.2). Secondly, they also include rules on the management of revocation information appearing in the `SignerRevReq` element (see clause 8.8.3).

The rules regarding use of Time Stamping Authorities appear in the `TimeStampTrustCondition` element (see clause 8.9).

The rules on the use of Attribute Authorities issuing Attribute Certificates appear in the `RoleTrustCondition` element (see clause 8.10).

The rules on the use of algorithms by the different present agents appear in the `AlgorithmConstraintSet` element (see clause 8.11).

Usually, an electronic signature produced under a security policy supports a number of commitments.

Some of the rules specified in a signature policy refer to the whole set of commitments made by the signer. The `CommonRules` element is specified in clause 8.4.

Other rules only apply to a certain given commitment. The `CommitmentRules` element is specified in clause 8.5 and the `RecognizedCommitmentType` element supporting the specification of the commitments themselves is specified in clause 8.6.2.

The present document specifies a formal structure in XML for an explicit Signature Validation Policy, and although other formats are allowed, for a given explicit signature there shall be one definitive form that has a unique binary encoded value.

Although the present document does not mandate the precise content of a signature policy, it has to be sufficiently definitive to avoid any ambiguity as to its implementation requirements. It shall be absolutely clear under which conditions an electronic signature should be accepted.

7 Namespace for this version

The XML namespace URI that must be used by implementations of this version of the present document is:

"http://uri.etsi.org/2038/v1.1.1"

The following namespace definitions will apply throughout all the present document:

```
<?xml version="1.0" encoding="UTF-8"?>

<xsd:schema xmlns:ds="http://www.w3.org/2000/02/xmldsig"
xmlns="http://uri.etsi.org/2038/v1.1.1#""
xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
xmlns:XAdES="http://uri.etsi.org/01903/v1.1.1#"
targetNamespace="http://uri.etsi.org/2038/v1.1.1#"
elementFormDefault="qualified">
```

8 Syntax Overview for Signature Policy

This clause contains the XML schema definitions for Signature Policies. These definitions are based on the information specified in TS 101 733 [1]. Each clause contains a rationale introducing the schema definition, the definition itself and additional textual explanations.

It is the opinion of the author that a much more mature document has to be produced by incorporating what RDF and P3P can offer, and in consequence, to get a final document fully ready to be part of the "Semantic Web". Such a document will be the result of future work that will be made available in successive versions of the present report.

8.1 The `SignaturePolicy` element

The root element for the XML signature policy specification is the `SignaturePolicy` element, whose XML schema definition is shown below:

```
<xsd:element name="SignaturePolicy"
 type="SignaturePolicyType"/>

<xsd:complexType name="SignaturePolicyType">
  <xsd:sequence>
    <xsd:element name="SignPolicyDigestAlg" type="ds:DigestMethodType"/>
    <xsd:element ref="ds:Transforms" minOccurs="0"/>
    <xsd:element name="SignPolicyInfo" type="SignPolicyInfoType"/>
    <xsd:element name="SignPolicyDigest" type="ds:DigestValueType"/>
    </xsd:sequence>
  </xsd:complexType>
```

The `SignPolicyInfo` element contains the computer processable information of the signature policy.

The `SignPolicyDigestAlg` element indicates the digest algorithm used to compute a digest value for the unique binary encoded value of the definitive form of the signature policy.

The optional `ds:Transforms` element can be used to specify a chain of transformations that has to be applied to the data before being digested. The processing model for the chain of transformations is as defined in clause 4.3.3.2 of [4].

The `SignPolicyDigest` element contains the aforementioned digest value. The signer shall include it so that it can be verified that the policy selected by the signer is identical to the one being used the verifier

`SignPolicyInfo` element is specified in clause 8.2.

8.2 The `SignPolicyInfo` element

The general information to be recorded about the signature policy should include:

Signature Policy Identifier: the "Signature Policy" will be identifiable by an identifier (`SignPolicyIdentifier` element).

Date of issue: when the "Signature Policy" was issued (`DateOfIssue` element).

Signature Policy Issuer name: an identifier for the body responsible for issuing the Signature Policy. This may be used by the signer or verifier in deciding if a policy is to be trusted, in which case the signer/verifier shall authenticate the origin of the signature policy as coming from the identified issuer (`PolicyIssuerName` element).

Field of application: this defines in general terms the general legal/contract/application contexts in which the signature policy is to be used and the specific purposes for which the electronic signature is to be applied (`FieldOfApplication` element).

Definition of the rules which form the **Signature Policy Validation** as described in subsequent clauses (`SignatureValidationPolicy` element). They are fully processable to allow the validation of electronic signatures issued under that signature policy.

Optionally, a set of extensions (`SignPolExtensions` element); whose definition is left open

Below follows the XML schema definition for this element.

```
<xsd:element name="SignPolInfo"
 type="SignaturePolicyInfoType"/>

<xsd:complexType name="SignaturePolicyInfoType">
  <xsd:sequence>
    <xsd:element name="SignPolicyIdentifier"
      type="XAdES:ObjectIdentifier"/>
    <xsd:element name="DateOfIssue" type="xsd:timeInstant"/>
    <xsd:element name="PolicyIssuerName" type="xsd:string"/>
    <xsd:element name="FieldOfApplication" type="xsd:string"/>
    <xsd:element name="SignatureValidationPolicy"
      type="SignatureValidationPolicyType"/>
    <xsd:element name="SignPolExtensions"
      type="SignPolExtensionsListType minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="SignPolExtensionsListType">
  <xsd:sequence maxOccurs="unbounded">
    <xsd:element name=SignPolExtension type="XAdES:AnyType"/>
  </xsd:sequence>
</xsd:complexType>
```

8.3 The `SignatureValidationPolicy` element

The signature validation policy defines a number of rules that have to be followed by both the signer when producing the electronic signature and by the verifier when verifying such an electronic signature. These rules refer to a number of different commitments being supported by electronic signatures produced under the security policy.

A signature validation policy should then specify:

A signing period, which identifies the date and time before which the signature policy should not be used for creating signatures, and an optional date after which it should not be used for creating signatures.(`SigningPeriod` element).

A list of rules to be applied to all the different commitment types (`CommonRules` element).

A list of specific rules that only apply to certain given commitment types (`CommitmentRules` element).

Optionally a number of qualifying extensions whose type is left open.

Below follow the XML schema definition for this element.

```
<xsd:element name="SignatureValidationPolicy"
 type="SignatureValidationPolicyType"/>

<xsd:complexType name="SignatureValidationPolicyType">
  <xsd:sequence>
    <xsd:element name="SigningPeriod" type="TimePeriodType"/>
    <xsd:element name="CommonRules" type="CommonRulesType"/>
    <xsd:element name="CommitmentRules" type="CommitmentRulesListType"/>
    <xsd:element name="SignPolicyExtensions" type="XAdES:AnyType"
      minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="TimePeriodType">
  <xsd:sequence>
    <xsd:element name="NotBefore" type="xsd:timeInstant"/>
    <xsd:element name="NotAfter" type="xsd:timeInstant" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
```

TS 101 733 [1] defines the ASN.1 CommitmentTypeIndication type and TS 101 903 [5] defines the XML CommitmentTypeIndicationType type. Both of them allow for the addition of information of the type of commitment got by producing an electronic signature of a certain data object. For any additional information on these types, please refer to these documents.

8.4 The CommonRules element

As it has been said before, the CommonRules element specifies rules that are common to all commitment types.

These rules are defined in terms of:

Rules for signer and verifier (SignerAndVerifier element).

Trust conditions for certificates (SigningCertTrustCondition element, see clause 8.8).

Trust conditions for timestamps (TimeStampTrustCondition element section, see clause 8.9).

Trust conditions for roles (RoleTrustCondition element section, see clause 8.10).

Constraints on algorithms (AlgorithmConstratintSet element section, see clause 8.11).

Below follow the XML schema definition for this element.

```
<xsd:element name="CommonRules"
 type="CommonRulesType"/>

<xsd:complexType name="CommonRulesType">
  <xsd:sequence>
    <xsd:element name="SignerAndVerifierRules"
     type="SignerAndVerifierRulesType" minOccurs="0"/>
    <xsd:element name="SigningCertTrustCondition"
     type="SigningCertTrustConditionType"
     minOccurs="0"/>
    <xsd:element name="TimeStampTrustCondition"
     type="TimeStampTrustCondition" minOccurs="0"/>
    <xsd:element name="RoleTrustCondition"
     type="RoleTrustConditionType" minOccurs="0"/>
    <xsd:element name="AlgorithmConstraintSet"
     type="AlgorithmConstraintSetType" minOccurs="0"/>
    <xsd:element name="SIgnPolExtensions"
     type="SignPolExtensionsListType" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="SignerAndVerifierRulesType">
  <xsd:sequence>
    <xsd:element name="SignerRules"
     type="SignerRulesType"/>
    <xsd:element name="VerifierRules"
     type="VerifierRulesType"/>
  </xsd:sequence>
</xsd:complexType>
```

If a field is present in CommonRules then the equivalent field shall not be present in any of the CommitmentRules (see below). If any of the following fields are not present in CommonRules then it shall be present in each CommitmentRule:

SignerAndVeriferRules;

SigningCertTrustCondition;

TimeStampTrustCondition.

8.5 The CommitmentRules element

As it has been said above, the CommitmentRules element specifies the validation rules that apply to given commitment types. Essentially it is a sequence where each element has the same contents as the CommonRules plus the SelCommitmentTypes element. As for the common rules, these rules are defined in terms of rules for signer and verifier and trust conditions for certificates, timestamps and roles, along with any constraints on algorithms.

Below follow the XML schema definition for this element.

```
<xsd:element name="CommitmentRules"
 type="CommitmentRulesListType"/>

<xsd:complexType name="CommitmentRulesListType">
   <xsd:sequence maxOccurs="unbounded">
      <xsd:element name="CommitmentRule" type="CommitmentRuleType"/>
   </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="CommitmentRuleType">
   <xsd:sequence>
      <xsd:element name="SelCommitmentTypes"
        type="SelectedCommitmentTypes"/>
      <xsd:element name="SignerAndVerifierRules"
        type="SignerAndVerifierRulesType" minOccurs="0"/>
      <xsd:element name="SigningCertTrustCondition"
        type="SigningCertTrustConditionType" minOccurs="0"/>
      <xsd:element name="TimeStampTrustCondition"
        type="TimeStampTrustConditionType" minOccurs="0"/>
      <xsd:element name="RoleTrustCondition"
        type="RoleTrustConditionType" minOccurs="0"/>
      <xsd:element name="AlgorithmConstraintSet"
        type="AlgorithmConstraintSetType" minOccurs="0"/>
      <xsd:element name="SignPolExtensions" type="SignPolExtensionsListType"
        minOccurs="0"/>
   </xsd:sequence>
</xsd:complexType>
```

8.6 Commitments elements

This clause shows the information related to the commitments taken by a certain agent under the signature policy being specified.

Clause 8.6.1 specifies the XML schema definition for the element containing the information on the commitments taken. Clause 8.6.2 specifies the XML schema definition for the semantics of each commitment taken.

8.6.1 The `SelCommitmentTypes` element

The `SelCommitmentTypes` element is used to indicate the commitment taken by a certain agent under the signature policy being specified.

Below follows the XML schema definition for this element:

```
<xsd:element name="SelCommitmentTypes"
 type="SelectedCommitmentTypeList"/>

<xsd:complexType name="SelectedCommitmentTypeList">
  <xsd:sequence maxOccurs="unbounded">
    <xsd:element name="SelCommitmentType"
     type="SelectedCommitmentType">
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="SelectedCommitmentType">
  <xsd:choice>
    <xsd:element name="Empty"/>
    <xsd:element name="RecognizedCommitmentType"
     type="CommitmentType"/>
  </xsd:choice>
</xsd:complexType>
```

It can be seen that this element contains a list of selected commitments whose semantic is given in the `RecognizedCommitmentType` elements.

If a certain `SelCommitmentType` contains an `Empty` element, it indicates that this rule is applied when a commitment type is not present in the electronic signature (i.e. the type of commitment is indicated in the semantics of the message). Otherwise, the electronic signature shall contain a commitment type indication that shall fit one of the commitments types that are mentioned in the `RecognizedCommitmentType` elements.

8.6.2 The `RecognizedCommitmentType` element

This element contains the semantic of each of the commitments taken by certain agents under the specified signature policy.

Below follows the XML schema definition for this element:

```
<xsd:element name="RecognizedCommitmentType"
 type="CommitmentType"/>

<xsd:complexType name="CommitmentType">
  <xsd:sequence>
    <xsd:element name="CommitmentIdentifier"
     type="XAdES:ObjectIdentifierType"/>
    <xsd:element name="FieldOfApplication"
     type="xsd:string" minOccurs="0"/>
    <xsd:element name="Semantics" type="xsd:string"
     minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
```

The `CommitmentIdentifier` element identifies the commitment being present in the signature policy.

The `FieldOfApplication` and `Semantics` elements define the specific use and meaning of the commitment within the overall field of application defined for the policy.

8.7 Rules on the signer and on the verifier

By specifying the requirements on the signer and verifier the responsibilities of the two parties can be clearly defined to establish all the necessary information.

These verification data rules should include:

requirements on the signer to provide given signed qualifying properties and roles;

requirements on the verifier to obtain additional certificates, CRLs, results of on line certificate status checks and to use timestamps (if no already provided by the signer).

8.7.1 The `SignerRules` element

The signer rules identify:

If the signed objects are external to the `Signature` element (`ExternalSignedObjects` element).

The signed qualifying properties (as specified in TS 101 903 [5]) that shall be provided by the signer under this policy (`MandatedSignedQProperties` element).

the unsigned qualifying properties (as specified in TS 101 903 [5]) that shall be provided by the signer under this policy (`MandatedUnsignedQProperties` element).

Whether the certificate identifiers from the full certification path up to the trust point shall be provided by the signer in the `SigningCertificate` qualifying property defined in TS 101 903 [5] (`MandatedCertificateRef` element).

Whether a signer's certificate, or all certificates in the certification path to the trust point shall be provided by the signer in the `KeyInfo` element of `Signature (MandatedCertificateInfo element)`.

Below follows the XML schema definition for this element:

```
<xsd:element name="SignerRules"
 type="SignerRulesType"/>

<xsd:complexType name="SignerRulesType">
  <xsd:sequence>
    <xsd:element name="ExternalSignedObjects"
     type="xsd:boolean" minOccurs="0"/>
    <xsd:element name="MandatedSignedQProperties"
     type="QPropertiesListType"/>
    <xsd:element name="MandatedUnsignedQProperties"
     type="QPropertiesListType"/>
    <xsd:element name="MandatedCertificateRef"
     type="CertificateReqType"/>
    <xsd:element name="MandatedCertificateInfo"
     type="CertificateReqType"/>
    <xsd:element name="SignPolicyExtensions"
     type="SignPolExtensionsListType" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="QPropertiesListType">
  <xsd:sequence maxOccurs="unbounded">
    <xsd:element name="QPropertyID"
     type="xsd:anyURI"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:simpleType name="CertificateReqType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="signerOnly"/>
    <xsd:enumeration value="fullPath"/>
  </xsd:restriction>
</xsd:simpleType>
```

The `MandatedSignedQProperties` element shall include the identifier for all those signed qualifying properties required by the present document as well as additional qualifying properties required by the signature policy.

The `MandatedUnsignedQProperties` element shall include the identifier for all those unsigned qualifying properties required by the present document as well as additional qualifying properties required the signature policy. For example, if a `SignatureTimestamp` element (whose XML schema definition appears in TS 101 903 [5]) is required *by the signer* the corresponding URI for this element shall be included.

The `MandatedCertificateRef` identifies whether just a reference to the signer's certificate, or references to the full certificate path shall be provided by the signer.

The `mandatedCertificateInfo` field identifies whether a signer's certificate, or all certificates in the certification path to the trust point shall be provided by the signer in the `KeyInfo` field of `Signature`.

8.7.2 The `VerifierRules` element

The verifier rules identify the unsigned qualifying properties that shall be present under this policy and shall be added to the electronic signature by the verifier if not added by the signer.

Below follows the XML schema for this element:

```
<xsd:element name="VerifierRules"
 type="VerifierRulesType"/>

<xsd:complexType name="VerifierRulesType">
  <xsd:sequence>
    <xsd:element name="MandatedQUnsignedProperties"
     type="QPropertiesListType"/>
    <xsd:element name="SignPolicyExtensions"
     type="SignPolExtensionsListType" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
```

`QpropertiesListType` type is defined in clause 8.7.1. `SignPolExtensionsListType` is defined in clause 8.2.

8.8 The `SigningCertTrustCondition` element

The `SigningCertTrustCondition` field identifies:

Trust conditions for certificate path processing used to validate the signing certificate (`SignerTrustTrees` element).

Minimum requirements for revocation information (`CertificateRevReq` element).

Below follows the XML schema definition for this element:

```
<xsd:element name="SigningCertTrustCondition"
 type="SigningCertTrustConditionType"/>

<xsd:complexType name="SigningCertTrustConditionType">
  <xsd:sequence>
    <xsd:element name="SignerTrustTrees"
     type="CertificateTrustTreesType"/>
    <xsd:element name="SignerRevReq"
     type="CertificateRevReqType"/>
  </xsd:sequence>
</xsd:complexType>
```

Clause 8.8.1 contains a detailed rationale on the conditions for the certificate path processing as appears in TS 101 733 [1]. Clause 8.8.2 contains the XML schema definition for the `SignerTrustTrees` elements incorporating the information concerning to the aforementioned conditions.

Clause 8.8.3 contains the rationale on the requirements for revocation information and the XML schema definition for the `SignerRevReq` element that incorporate information on these requirements.

8.8.1 Rules for use of Certification Authorities

The certificate validation process of the verifier, and hence the certificates that may be used by the signer for a valid electronic signature, may be constrained by the combination of the trust point and certificate path constraints in the signature validation policy.

8.8.1.1 Trust Points

The signature validation policy defines the certification authority trust points that are to be used for signature verification. Several trust points may be specified under one signature policy. Specific trust points may be specified for a particular type of commitment defined under the signature policy. For a signature to be valid a certification path shall exists between the Certification Authority that has granted the certificate selected by the signer (i.e. the used user-certificate) and one of the trust point of the Signature Validation Policy.

8.8.1.2 Certification Path

There may be constraints on the use of certificates issued by one or more CA(s) in the certificate chain and trust points. The two prime constraints are certificate policy constraints and naming constraints.

Certificate policy constraints limit the certification chain between the user certificate and the certificate of the trusted point to a given set of certificate policies, or equivalents identified through certificate policy mapping.

The naming constraints limit the forms of names that the CA is allowed to certify.

Name constraints are particularly important when a Signature policy identifies more than one trust point. In this case, a certificate of a particular trusted point may only be used to verify signatures from users with names permitted under the name constraint.

Certificate Authorities may be organized in a tree structure, this tree structure may represent the trust relationship between various CA(s) and the users CA. Alternatively, a mesh relationship may exist where a combination of tree and peer cross-certificates may be used. The requirement of the certificate path in the present document is that it provides the trust relationship between all the CAs and the signers user certificate. The starting point from a verification point of view, is the trust point. A trust point, usually a CA that publishes self-certified certificates, is the starting point from which the verifier verifies the certificate chain. Naming constraints may apply from the trust point, in which case they apply throughout the set of certificates that make up the certificate path down to the signer's user certificate.

Policy constraints can be easier to process but to be effective require the presence of a certificate policy identifier in the certificates used in a certification path.

Certificate path processing, thus generally starts with one of the trust point from the signature policy and ends with the user certificate.

The certificate path processing procedures defined in RFC 2560 [11] clause 6 identifies the following initial parameters that are selected by the verifier in certificate path processing:

acceptable certificate policies;

naming constraints in terms of constrained and excluded naming subtree;

requirements for explicit certificate policy indication and whether certificate policy mapping are allowed;

restrictions on the certificate path length.

The signature validation policy identifies constraints on these parameters in the Certificate

8.8.2 The SignerTrustTrees element

The SignerTrustTrees element identifies a set of self signed certificates for the trust points (CertificateTrustPoint elements) used to start (or end) certificate path processing and the initial conditions for certificate path validation as defined RFC 2459 [3] clause 6. As it has been said, this element is used to define policy for validating the signing certificate, the TSA's certificate and attribute certificates.

Below follows the XML schema definition of this element:

```
<xsd:element name="SignerTrustTrees" type="CertificateTrustTreesType"/>

<xsd:complexType name="CertificateTrustTreesType">
  <xsd:sequence maxOccurs="unbounded">
    <xsd:element name="CertificateTrustPoint"
     type="CertificateTrustPointType"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="CertificateTrustPointType">
  <xsd:sequence>
    <xsd:element name="TrustPoint"
     type="ds:X509CertificateType"/>
    <xsd:element name="PathLenConstraint"
     type="xsd:integer" minOccurs="0"/>
    <xsd:element name="AcceptablePolicySet"
     type="AcceptablePoliciesListType" minOccurs="0"/>
    <xsd:element name="NameConstraints"
     type="NameConstraintsType" minOccurs="0"/>
    <xsd:element name="PolicyConstraints"
     type="PolicyConstraintsType" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="AcceptablePoliciesListType">
  <xsd:sequence maxOccurs="unbounded">
    <xsd:element name="AcceptablePolicy"
     type="XAdES:ObjectIdentiferType"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="NameConstraintsType">
  <xsd:sequence>
    <xsd:element name="PermittedSubtrees"
     type="GeneralSubTreesListType" minOccurs="0"/>
    <xsd:element name="ExcludedSubtrees"
     type="GeneralSubTreesListType" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="GeneralSubTreesListType">
  <xsd:sequence maxOccurs="unbounded">
    <xsd:element name="GeneralSubTree" type="GeneralSubTreeType"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="GeneralSubTreeType">
  <xsd:sequence>
    <xsd:element name="Base" type="xsd:string"/>
    <xsd:element name="Minimum" type="xsd:integer" default="0"/>
    <xsd:element name="Maximum" type="xsd:integer" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="PolicyConstraintsType">
  <xsd:sequence>
    <xsd:element name="RequireExplicitPolicy" type="xsd:integer"
      minOccurs="0"/>
    <xsd:element name="InhibitExplicitPolicy" type="xsd:integer"
      minOccurs="0"/>
  </xsd:sequence>
```

```
</xsd:complexType>
```

The `TrustPoint` element gives the self signed certificate for the CA that is used as the trust point for the start of certificate path processing.

The `PathLenConstraint` element gives the maximum number of CA certificates that may be in a certification path following the trustpoint. A value of zero indicates that only the given trustpoint certificate and an end-entity certificate may be used. If present, the pathLenConstraint field shall be greater than or equal to zero. Where pathLenConstraint is not present, there is no limit to the allowed length of the certification path.

The `AcceptablePolicySet` element identifies the initial set of certificate policies, any of which are acceptable under the signature policy.

The `NameConstraints` field indicates a name space within which all subject names in subsequent certificates in a certification path shall be located. Restrictions may apply to the subject distinguished name or subject alternative names. Restrictions apply only when the specified name form is present. If no name of the type is in the certificate, the certificate is acceptable. These restrictions are defined in terms of permitted (`PermittedSubtrees element`) or excluded name subtrees (`ExcludedSubtrees` element). Any name matching a restriction in the `ExcludedSubtrees` element is invalid regardless of information appearing in the `PermittedSubtrees element`.

Finally, the `PolicyConstraints` element constrains path processing in two ways. It can be used to prohibit policy mapping or require that each certificate in a path contain an acceptable policy identifier. If present, this element specifies requirement for explicit indication of the certificate policy and/or the constraints on policy mapping.

If the `InhibitPolicyMapping` element is present within the `PolicyConstraints` element, the value indicates the number of additional certificates that may appear in the path (including the trustpoint's self certificate) before policy mapping is no longer permitted. For example, a value of one indicates that policy mapping may be processed in certificates issued by the subject of this certificate, but not in additional certificates in the path.

If the `RequireExplicitPolicy` element is present, subsequent certificates shall include an acceptable policy identifier. The value of `RequireExplicitPolicy` indicates the number of additional certificates that may appear in the path (including the trustpoint's self certificate) before an explicit policy is required. An acceptable policy identifier is the identifier of a policy required by the user of the certification path or the identifier of a policy that has been declared equivalent through policy mapping.

8.8.3 The `SignerRevReq` element

The signature policy should define rules specifying requirements for the use of Certificate Revocation Lists (CRLs) and/or on-line certificate status check service to check the validity of a certificate. These rules specify the mandated minimum checks that shall be carried out.

It is expected that in many cases either check may be selected with checks of CRLs being carried out for certificate status that are unavailable from OCSP servers. The verifier may take into account information in the certificate in deciding how best to check the revocation status (e.g. a certificate extension field about authority information access or a CRL distribution point) provided that it does not conflict with the signature policy revocation rules.

Below follows the XML schema definition for this element:

```
<xsd:element name="SignerRevReq" type="CertificateRevReqType"/>

<xsd:complexType name="CertificateRevReqType">
  <xsd:sequence>
    <xsd:element name="EndRevReq" type="RevocationReqType"/>
    <xsd:element name="CACerts" type="RevocationReqType"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="RevocationReqType">
  <xsd:sequence>
    <xsd:element name="EnuRevReq"
     type="EnuRevReqType"/>
    <xsd:element name="exRevReq" type="SignPolExtensionsListType"
      minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:simpleType name="EnuRevReqType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="clrcheck"/>
    <xsd:enumeration value="ocspcheck"/>
    <xsd:enumeration value="bothcheck"/>
    <xsd:enumeration value="eithercheck"/>
    <xsd:enumeration value="nocheck"/>
    <xsd:enumeration value="other"/>
  </xsd:restriction>
</xsd:simpleType>
```

Certificate revocation requirements are specified in terms of checks required on:

End certificates (i.e. the signers certificate, the attribute certificate or the timestamping authority certificate). These requirements appear in the `EndCertRevReq element`.

CA certificates. These requirements appear in the `CACerts element`.

Revocation requirements are specified in terms of:

`clrCheck`: Checks shall be made against current CRLs (or authority revocation lists);

`ocspCheck`: The revocation status shall be checked using the Online Certificate Status Protocol (RFC 2450 [10]);

`bothCheck`: Both OCSP and CRL checks shall be carried out;

`eitherCheck`: Either OCSP or CRL checks shall be carried out;

`noCheck`: No check is mandated.

8.9 The `TimeStampTrustCondition` element

The `TimeStampTrustCondition` element identifies trust conditions for certificate path processing used to authenticate the timestamping authority and constraints on the name of the timestamping authority. This applies to the timestamp that shall be present in every XAdES-T electronic signature format as defined in TS 101 903 [5].

The following rules should be used when specifying, constraints on the certificate paths for timestamping authorities, constraints on the timestamping authority names and general timing constraints:

Trust Points and Certificate Paths. Signature keys from timestamping authorities will need to be supported by a certification path. The certification path used for timestamping authorities requires a trust point and possibly path constraints in the same way that the certificate path for the signer's key.

Timestamping Authority Names. Restrictions may need to be placed by the validation policy on the named entities that may act a timestamping authorities.

Timing Constraints - Caution Period. Before an electronic signature may really be valid, the verifier has to be sure that the holder of the private key was really the only one in possession of key at the time of signing. However, there is an inevitable delay between a compromise or loss of key being noted, and a report of revocation being distributed. To allow greater confidence in the validity of a signature, a "cautionary period" may be identified before a signature may be said to be valid with high confidence. A verifier may revalidate a signature after this cautionary signature, or wait for this period before validating a signature. The validation policy may specify such a cautionary period.

Timing Constraints - Timestamp Delay. There will be some delay between the time that a signature is created and the time the signer's digital signature is timestamped. However, the longer this elapsed period the greater the risk of the signature being invalidated due to compromise or deliberate revocation of its private signing key by the signer. Thus the signature policy should specify a maximum acceptable delay between the signing time as claimed by the signer and the time included within the timestamp.

Below follows the XML schema definition for this element:

```
<xsd:element name="TimeStampTrustCondition"
 type="TimeStampTrustConditionType"/>

<xsd:complexType name="TimeStampTrustConditionType">
  <xsd:sequence>
    <xsd:element name="TtsCertificateTrustTrees"
     type="CertificateTrustTreesType" minOccurs="0"/>
    <xsd:element name="TtsRevReq" type="CertificateRevReqType"
      minOccurs="0"/>
    <xsd:element name="TtsNameConstraints" type="NameConstraintsType"
      minOccurs="0"/>
    <xsd:element name="CautionPeriod" type="DeltaTimeType" minOccurs="0"/>
    <xsd:element name="SignatureTimeStampDelay" type="DeltaTimeType"
      minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="DeltaTimeType">
  <xsd:sequence>
    <xsd:element name="DeltaSeconds" type="xsd:integer"/>
    <xsd:element name="DeltaMinutes" type="xsd:integer"/>
    <xsd:element name="DeltaHours" type="xsd:integer"/>
    <xsd:element name="DeltaDays" type="xsd:integer"/>
  </xsd:sequence>
</xsd:complexType>
```

`If TtsCertificateTrustTrees` element is not present then the same rule as defined in `SigningCertTrustCondition` element applies to certification of the timestamping authorities public key.

The `TsRevReq` element specifies minimum requirements for revocation information, obtained through CRLs and/or OCSP responses, to be used in checking the revocation status of the time stamp that shall be present in the XAdES-T.

If `TtsNameConstraints` is not present then there are no additional naming constraints on the trusted timestamping authority other than those implied by the `TtsCertificateTrustTrees` element.

The `CautionPeriod` element specifies a caution period after the signing time that it is mandated the verifier shall wait to get high assurance of the validity of the signer's key and that any relevant revocation has been notified. The revocation status information forming the ES with Complete validation data shall not be collected and used to validate the electronic signature until after this caution period.

The `SignatureTimestampDelay` element specifies a maximum acceptable time between the signing time and the time at which the signature timestamp, as used to form the ES Timestamped, is created for the verifier. If the signature timestamp is later that the time in the signing-time attribute by more than the value given in `SignatureTimestampDelay`, the signature shall be considered invalid.

8.10 The `RoleTrustCondition` element

Roles can be supported as claimed roles or as certified roles using Attribute Certificates. The following rules should be in the management of roles:

Role Values. When signature under a role is mandated by the signature policy, then either Attribute Certificates may be used or the signer may provide a claimed role. The acceptable role types or values may be dependent on the type of commitment. For example, a user may have several roles that allow the user to sign data that imply commitments based on one or more of his roles.

Trust Points for Certified Attributes. When a signature under a certified role is mandated by the signature policy, Attribute Authorities -AA(s)- (Authorities that issue Attribute Certificates) are used and need to be validated as part of the overall validation of the electronic signature. The trust points for Attribute Authorities do not need to be the same as the trust points to evaluate a certificate from the CA of the signer. Thus the trust point for verifying roles need not be the same as trust point used to validate the certificate path of the user's key. Naming and certification policy constraints may apply to the AA in similar circumstance to when they apply to CA. Constraints on the AA and CA need not be exactly the same. AA(s) may be used when a signer is creating a signature on behalf of an organization, they can be particularly useful when the signature represents an organizational role. AA(s) may or may not be the same authority as CA(s). Thus, the Signature Policy identifies trust points that can be used for Attribute Authorities, either by reference to the same trust points as used for Certification Authorities, or by an independent list.

Certification Path for Certified Attributes. Attribute Authorities may be organized in a tree structure in similar way to CAs, where the AAs are the leaves of such a tree. Naming and other constraints may be required on attribute certificate paths in a similar manner to other electronic signature certificate paths. Thus, the Signature Policy identifies constraints on the following parameters used as input to the certificate path processing:

acceptable certificate policies, including requirements for explicit certificate policy indication and whether certificate policy mapping is allowed;

naming constraints in terms of constrained and excluded naming subtrees;

restrictions on the certificate path length.

Below follows the XML schema definition for this element:

```
<xsd:element name="RoleTrustCondition" type="RoleTrustConditionType"/>

<xsd:complexType name="RoleTrustConditionType">
  <xsd:sequence>
    <xsd:element name="RoleMandated" type="xsd:boolean"/>
    <xsd:element name="HowCertRole" type="HowCertRoleType"/>
    <xsd:element name="AttrCertTrustTrees"
      type="CertificateTrustTreesType" minOccurs="0"/>
    <xsd:element name="RoleRevReq" type="CertificateRevReqType"
      minOccurs="0"/>
    <xsd:element name="RoleConstraints" type="RoleConstraintsType"
      minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:simpleType name="HowCertRoleType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="ClaimedRole"/>
    <xsd:enumeration value="CertifiedRole"/>
    <xsd:enumeration value="Either"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="RoleConstraintsType">
  <xsd:sequence >
    <xsd:element name="RoleTypeConstraint"
      type="XAdES:ObjectIdentifierType" minOccurs="0"
      maxOccurs="unbounded"/>
    <xsd:element name="RoleValueConstraint" type="XAdES:AnyType"
      minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
```

```
</xsd:complexType>
```

If the `RoleTrustCondition` element is not present then any certified roles within an attribute certificate may not considered to be valid under the validation policy.

If `RoleMandated` is true then a role, certified within the following constraints, shall be present. If false, then the signature is still valid if no role is specified.

The `HowCertRole` element specifies how must appear the roles within an electronic signature: uncertified roles "claimed" by the signer, or certified roles in an attribute certificate or either.

The `AttrCertTrustTrees` element specifies certificate path conditions for any attribute certificate. If not present the same rules apply as in `SigningCertTrustCondition`.

The `RoleRevReq` element specifies minimum requirements for revocation information, obtained through CRLs and/or OCSP responses, to be used in checking the revocation status of Attribute Certificates, if any are present.

If the `RoleConstraints` field is not present then there are no constraints on the roles that may be validated under this policy.

If a `RoleTypeConstraint` element is present within the `RoleConstraints` element, it specifies a role type that is considered valid under the signature policy. Any value for that role is considered valid.

If a `RoleValueConstraint` is present within the `RoleConstraints` element, it specifies a specific role value that is considered valid under the signature policy.

8.11 The `AlgorithmConstraintSet` element

The `AlgorithmConstrainSet` element, if present, identifies the signing algorithms (hash, public key cryptography, combined hash and public key cryptography) that may be used for specific purposes and any minimum length. If this element is not present then the policy applies no constraints.

Below follows the XML schema definition for this element:

```
<xsd:element name="AlgorithmConstraintSet"
 type="AlgorithmConstraintSetType"/>

<xsd:complexType name="AlgorithmConstraintSetType">
  <xsd:sequence>
    <xsd:element name="SignerAlgConstraints"
      type="AlgConstraintsListType" minOccurs="0"/>
    <xsd:element name="EeCertAlgConstraints"
     type="AlgConstraintsListType" minOccurs="0"/>
    <xsd:element name="CACertAlgConstraints"
     type="AlgConstraintsListType" minOccurs="0"/>
    <xsd:element name="AaCertAlgConstraints"
      type="AlgConstraintsListType" minOccurs="0"/>
    <xsd:element name="TSACertAlgConstraints"
      type="AlgConstraintsListType" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="AlgConstraintsListType">
  <xsd:sequence maxOccurs="unbounded">
    <xsd:element name="AlgAndLength" type="AlgAndLengthType"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="AlgAndLengthType">
  <xsd:sequence>
    <xsd:element name="AlgId" type="xsd:anyUri"/>
    <xsd:element name="MinKeyLength" type="xsd:integer" minOccurs="0"/>
    <xsd:element name="Other" type="SignPolExtensionsListType"
      minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
```

Using this XML schema definition, the signature validation policy may identify a set of signing algorithms (hashing, public key, combinations) and minimum key lengths that may be used:

by the signer in creating the signature (`SignerAlgConstraints` element).

in end entity public key Certificates (`EeCertAlgConstraints` element).

in CA Certificates (`CACertAlgConstraints` element).

in attribute Certificates (`AaCertAlgConstraints` element).

by the timestamping authority (`TSACertAlgConstraints` element).

The `MinKeyLength` element specifies the minimum length of the corresponding keys in bits.

9 Initial Comments on the potential improvements

The author envisages different types of potential improvements in the view of what it is offered by RDF and P3P:

In order to introduce Signature Policies within the "Semantic Web", it could be envisaged to produce a XML resource model encoded using RDF syntax as described in [6] and [7]. This model should be able to notify the properties on such a policy in a way that could for instance, allow agents in the Web to identify the policy as candidate to satisfy the requirements of certain communities. This would be a first way of taking benefit of the powerfulness of the RDF.

It should be explored the possibility of even introduce semantic descriptions within the Signature Policy XML document itself. New XML elements could, in consequence, be specified and incorporated to the `SignaturePolicy` element.

It should be taken into account that P3P has defined an exhaustive way of identifying legal entities, and that the issuers of Signature Policy will be legal entities. This puts on the table the issue of re-using the data structures for, at least this data element in the XML Signature Policy definition. Indeed, other data elements could be also taken into account to incorporate additional information on the document itself. At a very first glance, two immediate elements specified in P3P could be useful for the specification of a Signature Policy: an XML element able to contain detailed information on the legal entity issuing the policy, and an XML element able to contain information on dispute resolution.

Finally, as said before, the possibility of incorporating mechanisms defined in P3P to define new structured and unstructured data elements, as a way of creating new data related with Signature Policies, should be taken into account.

Annex A: Bibliography

Directive 1999/93/EC of the European Parliament and of the Council of 13 December 1999 on a Community framework for electronic signatures.

W3C Recommendation: "XML Schema Part 1: Structures".

W3C Recommendation: "XML Schema Part 2: Datatypes".

RFC 2634 (June 1999): "Enhanced Security Services for S/MIME".

ITU-T Recommendation X.509: "Information technology - Open Systems Interconnection - The directory: Public-key and attribute certificate frameworks".

ETSI TS 101 861: "Time stamping profile".

History

Document history		
V1.1.1	April 2002	Publication