



Question(s): L, M/17

Geneva, 10-19 March 2004

Ref. : TD 2400 and 3213**Source:** ITU-T Study Group 17, Geneva, 10 - 19 March 2004**Title:** Web Services Security and use of ASN.1 in SAML and XACML

LIAISON STATEMENT**To:** OASIS**Approval:** Agreed to at the ITU-T SG 17 meeting**For:** Action

Contacts: Dr. Heungyoul Youm
Rapporteur Q.L/17Tel: +82 41 530 1328
Fax: +82 41 530 1494
Email: hyyoum@sch.ac.krJohn Larmouth
Rapporteur Q.M/17Tel: +44 161 928 1605
Fax: +44 161 928 8069
Email: j.larmouth@salford.ac.uk

ITU-T Study Group 17 would like to thank OASIS for submitting SAML v1.1 and XACML v1.0 as proposal for ITU-T Recommendations. We also greatly appreciated the direct participation of OASIS leadership in our meeting and the informative tutorials that were given on OASIS, SAML and XACML.

The documents submitted were considered mainly within the Questions E/17 (Directory services, directory systems and public-key/attribute certificates), L/17 (Secure communication services) and M/17 (Abstract Syntax Notation One (ASN.1) and other data languages). The Directory group identified no issues with the documents from OASIS. Questions L/17 and M/17 prepared comments/proposals to OASIS that are presented below.

1 Q.L/17 (Secure communication services)

Q.L/17 recognizes that OASIS standards would have direct relationship with future activities on secure communication services. Q.L/17 wishes to collaborate closely with OASIS in order to develop new ITU Recommendations on web services security for telecommunications.

Although Q.L/17 has not consented on the submitted standards at this Study Group 17 meeting, Q.L/17 kindly ask OASIS to submit SAML v2.0 and XACML v2.0 during the next study period (2005-2008) in order to progress their approval as ITU-T Recommendations.

Q.L/17 plans to study web services security during the 2005-2008 study period as reflected in the Q.L/17 action plan (see Attachment 1).

Attention: Some or all of the material attached to this liaison statement may be subject to ITU copyright. In such a case this will be indicated in the individual document.

Such a copyright does not prevent the use of the material for its intended purpose, but it prevents the reproduction of all or part of it in a publication without the authorization of ITU.

2 Q.M/17 (Abstract Syntax Notation One (ASN.1) and other data languages)

2.1 Background

2.1.1 The ASN.1 group in ITU-T Study Group 17 considered TD 0221 "*OASIS SAML v1.1 standard, candidate new ITU-T Recommendation*" and TD 0222 "*OASIS XACML v1.0 standard, candidate new ITU-T Recommendation*", and made the following observations.

2.1.2 Early experiments by OSS Nokalva and other companies showed that a good integration of binary encodings (using the ITU-T language ASN.1 and its Packed Encoding Rules¹) with XML encodings was possible, and that this conferred significant advantages for telecommunications applications. They showed that ASN.1 should be in the picture for high-throughput transaction processing systems, low-bandwidth communications and low-power processors with small memory where compression (zipping) is not the answer.

2.1.3 Reasons for using ASN.1 binary encodings are:

- No tree information is stored in the binary encoding.
- On average the compression rate of binary encodings is better than that achieved by tools like zip.
- High-throughput transaction processing systems, low-bandwidth communications and low-power processors with small memory are not places where compression algorithms are usually recommended.
- When using ASN.1 binary encodings, the application is not slowed down because there is no need to build a dictionary on the fly. The encoders and decoders have been generated once for all.

2.1.4 ASN.1+PER has already been chosen as an alternative to XML in standards such as OASIS XCBF, ITU-T Rec. F.515 (*Unified Directory Specification*), ITU-T Rec. X.fws (*Fast Web Services*), and ITU-T Rec. X.finf (*Fast Infoset*).

2.1.5 For these gains to be fully realized, it is considered important that the approach taken by Q.M/17 for Fast Web Services and Fast Infoset be mirrored in any ITU-T Recommendations for SAML and XACML.

2.2 Proposals from the ITU-T ASN.1 group

2.2.1 There was discussion with the OASIS representatives in Geneva on the possibility of joint work, immediately following the production of version 2.0 of the OASIS standards, to enhance the texts with an ASN.1 specification. The ASN.1 specification would generate identical XML encodings to those in the base text, but would also allow binary encodings, making the result much more useful for telecommunications applications, and in particular with the ITU-T Recommendation for Fast Web Services.

2.2.2 ITU-T Study Group 17 would wish this work to be conducted on an equal basis between the two organizations, leading to common text to be published as both an OASIS standard and an ITU-T Recommendation.

2.2.3 OASIS is requested to confirm that the experts in the SAML and XACML TCs are prepared to work in collaboration with the ITU-T experts towards this goal.

2.2.4 The following Attachment 2 provides comparative results for tests that have been run and provides ASN.1 modules equivalent to the XML Schemas specified in the current OASIS SAML and XACML standards. The ASN.1 modules (currently lacking supporting text) provide an illustration of what would be added to SAML 2.0 and XACML 2.0.

¹ The Packed Encoding Rules (PER) of ASN.1 are defined in [ITU-T Rec. X.691 | ISO/IEC 8825-2](#).

Attachment 1

Action plan for Question Q.L/17

1. Question Q.L/17: Secure Communication Services

2. Objectives

Mobile security

- Identify threats behind mobile communications services, and set up technical countermeasures to handle them
- Study a general security policy technologies for the end-to-end mobile communications
- Study a security infrastructure, secure application protocols, and comprehensive security solutions in the emerging mobile network like ubiquitous environment
- Maintain the existing mobile security-related Recommendations, produce and consent new ITU-T Recommendations for the end-to-end mobile communications

Secure application protocols

- Study and recommend various secure application protocols such as PKI application protocol, end-to-end transport protocol, secure key exchange protocol, password authentication protocol, authorization protocol, notarisation protocol, and time stamping protocol, et al, in a secure communication service.
- Standardize secure communication services to support secure and stable operation of the telecommunication network and improve the quality of security in a communication network.
- Standardize secure interconnectivity methods for secure application services.
- Standardize the security mechanisms for supporting secure application services
- Produce and consent new ITU-T Recommendations for secure application protocols

Web services security

- Investigate web services security technologies including security authentication assertion and extensible access control assertion, single-sign-on, et al.
- Clarify secure web services within Telecommunications architectures; work out the usage of web services security for Telecommunication scenarios, networks, its systems and related applications.
- Recommend guidelines how to best protect web services telecommunication systems from attacks.
- Coordinate web services security activities with OASIS and/or other groups.
- Produce and consent new ITU-T Recommendations for web services security.

Other relevant topics

- Study and develop security protocol interoperability test.
- Produce and consent new ITU-T Recommendations for security protocol interoperability test

3. Existing Recommendations

No.	Title
X.1121	Framework of secure technologies for mobile end-to-end data communication
X.1122	Guideline for implementing secure mobile systems based on PKI

4. Liaisons

- ITU-T SGs 2, 4, 9, 11, 13, 16, SSG; ITU-R
- ISO/IEC JTC 1/SC 27
- OASIS
- IETF
- ETSI
- OMA
- Liberty Alliance Project
- W3C
- ATIS
- 3GPP
- 3GPP2

5. Work programme and milestones

Draft and finalize list of items to be standardized in Q.L/17 1Q2005

First Draft of Recommendation X.websec-1	1Q/2005
First Draft of Recommendation X.websec-2	1Q/2005
First Draft of Recommendation X.msec-3	3Q/2005
First Draft of Recommendation X.secpro	3Q/2005
First Draft of Recommendation X.msec-4	1Q/2006
First draft amendment of Recommendation X.1121	1Q/2006
First draft amendment of Recommendation X.1122	1Q/2006
Final Draft of Recommendation X.websec-1	3Q/2006

Final Draft of Recommendation X.websec-2	3Q/2006
Final Draft of Recommendation X.msec-3	3Q/2007
Final Draft of Recommendation X.msec-4	1Q/2008
Final Draft of Recommendation X.secpro	1Q/2008

6. Future meetings

No interim meetings scheduled as yet. It is expected to have a joint interim meeting with other Questions, which are related to security technology in ITU-T.


7. Contact

Dr. H.Y. Youm
Dept. of Information Security Engineering
Soonchunhyang University
Asan-si, Choongnam-do
Republic of Korea

Tel: +82 41 530 1328
Fax: +82 41 530 1494
Email: hyyoum@sch.ac.kr

Attachment 2

SAML test results

A set of tests was run using the files listed in the first column of the following table. (The files are provided in this zipped archive:  SAML test1.zip .)

The pinked column gives the size of the original XML message. The blueed columns show the size of each message encoded in Aligned PER (APER), encoded in Unaligned PER (UPER) and compressed with zip respectively. Each yellowed column gives the ratio between the size of the binary encoding (in the blueed cell to the left of each yellowed cell) and the size of the original XML message.

Message	XML	APER	APER/XML	UPER	UPER/XML	WinZip	WinZip/XML
fullResponse	5493	2274	0.41	2219	0.40	1930	0.35
no-signatureResponse	1239	468	0.38	425	0.34	518	0.42
assertionArtifactRequest	363	136	0.37	123	0.34	268	0.74
assertionArtifactResponse	998	391	0.39	366	0.37	488	0.49
authenticationQueryRequest	578	203	0.35	190	0.33	354	0.61
authenticationQueryResponse	995	390	0.39	365	0.37	491	0.49
auth-assertion	823	332	0.40	300	0.36	430	0.52
testResponse	1467	512	0.35	463	0.32	603	0.41
authorizationDecisionQueryRequest	1275	485	0.38	470	0.37	515	0.40
authorizationDecisionQueryResponse	1479	649	0.44	630	0.43	621	0.42
attributeQueryRequest	1658	593	0.36	577	0.35	565	0.34
attributeQueryResponse	4984	1778	0.36	1761	0.35	899	0.18

In two-thirds of the cases, the most compact encodings are achieved by using Unaligned PER. In case the device (or equipment) is able to execute a compression software such as zip, the application of zip to the Aligned PER encodings achieves a better overall compression rate than the use of either PER or zip alone.


We have also encoded two of the messages (namely, "fullResponse" and "no-signatureResponse") using two variants of the Basic Encoding Rules² (BER) of ASN.1 (namely, definite length and indefinite length). We provide the results here as a matter of comparison, but we are not advocating the use of BER as a binary encoding for XML.

Message	XML	BER definite	BER def/XML	BER indefinite	BER indef/XML
fullResponse	5493	2470	0.45	2574	0.47
no-signatureResponse	1239	522	0.42	554	0.45

In the second example (*i.e.*, without the signatures), Unaligned PER is one third the size of the original XML message.

² The Basic Encoding Rules (BER) of ASN.1 are defined in [ITU-T Rec. X.690 | ISO/IEC 8825-1](http://www.itu-t.org/rec/T-REC-X.690).

XACML test results

A set of tests was run using the files listed in the first column of the following table. (The files are provided in this zipped archive:  XACML test.zip .)

For XACML, we have also measured the sizes of the Aligned-PER messages compressed with WinZip, which are usually better than either PER alone or WinZip alone (except in one case).

Message	XML	APER	APER/ XML	UPER	UPER/ XML	WinZip	WinZip/ XML	APERZ ³	APERZ/ APER	APERZ/ XML
policy-generated	3096	1372	0,44	1360	0,44	897	0,29	535	0,39	0,17
policy-obligation	3041	1348	0,44	1337	0,44	874	0,29	514	0,38	0,17
policy-selector	2341	1031	0,44	1024	0,44	772	0,33	446	0,43	0,19
time-range	3088	1313	0,43	1300	0,42	899	0,29	528	0,40	0,17
request-door-access	997	331	0,33	327	0,33	337	0,34	171	0,52	0,17
request-generated	1233	435	0,35	430	0,35	386	0,31	214	0,49	0,17
request-resource-content	1040	803	0,77	798	0,77	356	0,34	368	0,46	0,35
request-sensitive	1540	374	0,24	370	0,24	504	0,33	193	0,52	0,13

³ WinZipped Aligned PER

SAML ASN.1 modules

The following ASN.1 modules were automatically translated from the XML Schemas defined in OASIS SAML, OASIS XACML and W3C XML Signature by applying the mapping defined in [ITU-T Rec. X.694 | ISO/IEC 8825-5](#) "Mapping W3C XML Schema Definitions into ASN.1".

The information between square brackets "[.]" is called an XER encoding instruction. The syntax and semantics of XER encoding instructions is defined in [ITU-T Rec. X.693/Amd. 1 | ISO/IEC 8825-4/Amd. 1](#). Their purpose is to generate the same XML encoding as the XML Schema when the ASN.1 schemas are used together with the XML Encoding Rules⁴ (XER) of ASN.1. They have no influence on a binary encoding such as PER.

```
/* oasis-sstc-saml-schema-assertion-1.1 */
Assertion
DEFINITIONS AUTOMATIC TAGS ::=
BEGIN
IMPORTS
    NCName, ID, DateTime, String, AnyURI, AnyType, QName
    FROM XSD /* Module standardized in ITU-T Rec. X.694 | ISO/IEC 8825-5 */
    Signature, KeyInfo
    FROM Xmlldsig; /* Module defined below */

DecisionType ::= ENUMERATED {
    deny, indeterminate, permit}

AssertionIDReference ::= XSD.NCName

Assertion ::= AssertionType

AssertionType ::= SEQUENCE {
    assertionID XSD.ID,
    issueInstant XSD.DateTime,
    issuer XSD.String,
    majorVersion INTEGER,
    minorVersion INTEGER,
    conditions Conditions OPTIONAL,
    advice Advice OPTIONAL,
```

⁴ The XML Encoding Rules (XER) of ASN.1 are defined in [ITU-T Rec. X.693 | ISO/IEC 8825-4](#).


```
choice-list SEQUENCE (SIZE(1..MAX)) OF CHOICE {  
    statement Statement,  
    subjectStatement SubjectStatement,  
    authenticationStatement AuthenticationStatement,  
    authorizationDecisionStatement AuthorizationDecisionStatement,  
    attributeStatement AttributeStatement},  
signature Signature OPTIONAL}
```

Conditions ::= ConditionsType

```
ConditionsType ::= SEQUENCE {  
    notBefore XSD.DateTime OPTIONAL,  
    notOnOrAfter XSD.DateTime OPTIONAL,  
    choice-list SEQUENCE OF CHOICE {  
        audienceRestrictionCondition AudienceRestrictionCondition,  
        doNotCacheCondition DoNotCacheCondition,  
        condition Condition}}
```

Condition ::= ConditionAbstractType-derivations

ConditionAbstractType ::= SEQUENCE { }

AudienceRestrictionCondition ::= AudienceRestrictionConditionType

```
AudienceRestrictionConditionType ::= SEQUENCE {  
    audience-list SEQUENCE (SIZE(1..MAX)) OF audience Audience}
```

Audience ::= XSD.AnyURI

DoNotCacheCondition ::= DoNotCacheConditionType

DoNotCacheConditionType ::= SEQUENCE { }

Advice ::= AdviceType

```
AdviceType ::= SEQUENCE {  
    choice-list SEQUENCE OF CHOICE {  
        assertionIDReference AssertionIDReference,
```

```
assertion      Assertion,  
elem           UTF8String}}
```

Statement ::= StatementAbstractType-derivations

StatementAbstractType ::= SEQUENCE { }

SubjectStatement ::= SubjectStatementAbstractType-derivations

SubjectStatementAbstractType ::= SEQUENCE {
 subject Subject}

Subject ::= SubjectType

SubjectType ::= SEQUENCE {
 choice CHOICE {
 sequence SEQUENCE {
 nameIdentifier NameIdentifier,
 subjectConfirmation SubjectConfirmation OPTIONAL},
 subjectConfirmation SubjectConfirmation}}

NameIdentifier ::= NameIdentifierType

NameIdentifierType ::= SEQUENCE {
 format XSD.AnyURI OPTIONAL,
 nameQualifier XSD.String OPTIONAL,
 base XSD.String}

SubjectConfirmation ::= SubjectConfirmationType

SubjectConfirmationType ::= SEQUENCE {
 confirmationMethod-list SEQUENCE (SIZE(1..MAX)) OF
 confirmationMethod ConfirmationMethod,
 subjectConfirmationData SubjectConfirmationData OPTIONAL,
 keyInfo KeyInfo OPTIONAL}

SubjectConfirmationData ::= XSD.AnyType

ConfirmationMethod ::= XSD.AnyURI

AuthenticationStatement ::= AuthenticationStatementType

AuthenticationStatementType ::= SEQUENCE {
 authenticationInstant XSD.DateTime,
 authenticationMethod XSD.AnyURI,
 subject Subject,
 subjectLocality SubjectLocality OPTIONAL,
 authorityBinding-list SEQUENCE OF authorityBinding AuthorityBinding}

SubjectLocality ::= SubjectLocalityType

SubjectLocalityType ::= SEQUENCE {
 dnsAddress XSD.String OPTIONAL,
 ipAddress XSD.String OPTIONAL}

AuthorityBinding ::= AuthorityBindingType

AuthorityBindingType ::= SEQUENCE {
 authorityKind XSD.QName,
 binding XSD.AnyURI,
 location XSD.AnyURI}

AuthorizationDecisionStatement ::= AuthorizationDecisionStatementType

AuthorizationDecisionStatementType ::= SEQUENCE {
 decision DecisionType,
 resource XSD.AnyURI,
 subject Subject,
 action-list SEQUENCE (SIZE(1..MAX)) OF action Action,
 evidence Evidence OPTIONAL}

Action ::= ActionType

ActionType ::= SEQUENCE {
 namespace XSD.AnyURI OPTIONAL,
 base XSD.String

}

Evidence ::= EvidenceType

EvidenceType ::= SEQUENCE {
 choice-list SEQUENCE (SIZE(1..MAX)) OF CHOICE {
 assertionIDReference AssertionIDReference,
 assertion Assertion}}

AttributeStatement ::= AttributeStatementType

AttributeStatementType ::= SEQUENCE {
 subject Subject,
 attribute-list SEQUENCE (SIZE(1..MAX)) OF attribute Attribute}

AttributeDesignator ::= AttributeDesignatorType-derivations

AttributeDesignatorType ::= SEQUENCE {
 attributeName XSD.String,
 attributeNamespace XSD.AnyURI}

Attribute ::= AttributeType

AttributeType ::= SEQUENCE {
 attributeName XSD.String,
 attributeNamespace XSD.AnyURI,
 attributeValue-list SEQUENCE (SIZE(1..MAX)) OF
 attributeValue AttributeValue}

AttributeValue ::= XSD.AnyType

AttributeDesignatorType-derivations ::= CHOICE {
 attributeDesignatorType AttributeDesignatorType,
 attributeType AttributeType}

ConditionAbstractType-derivations ::= CHOICE {
 conditionAbstractType ConditionAbstractType,
 audienceRestrictionConditionType AudienceRestrictionConditionType,

doNotCacheConditionType DoNotCacheConditionType}

```
StatementAbstractType-derivations ::= CHOICE {
    statementAbstractType StatementAbstractType,
    attributeStatementType AttributeStatementType,
    authenticationStatementType AuthenticationStatementType,
    authorizationDecisionStatementType AuthorizationDecisionStatementType,
    subjectStatementAbstractType SubjectStatementAbstractType}
```

```
SubjectStatementAbstractType-derivations ::= CHOICE {
    subjectStatementAbstractType SubjectStatementAbstractType,
    attributeStatementType AttributeStatementType,
    authenticationStatementType AuthenticationStatementType,
    authorizationDecisionStatementType AuthorizationDecisionStatementType}
```

END

/ oasis-sstc-saml-schema-protocol-1.1 */*

Protocol

DEFINITIONS XER INSTRUCTIONS AUTOMATIC TAGS ::=

BEGIN

IMPORTS

DateTime, ID, QName, String, AnyURI, NCName

FROM XSD */* Module standardized in [ITU-T Rec. X.694 | ISO/IEC 8825-5](#) */*

Signature

FROM Xmlldsig */* Module defined below */*

AssertionIDReference, Subject, AttributeDesignator, Action, Evidence,

Assertion

FROM Assertion; */* Module defined above */*

RequestAbstractType ::= SEQUENCE {

issueInstant [NAME AS CAPITALIZED] [ATTRIBUTE] XSD.DateTime,

majorVersion [NAME AS CAPITALIZED] [ATTRIBUTE] INTEGER,

minorVersion [NAME AS CAPITALIZED] [ATTRIBUTE] INTEGER,

requestID [NAME AS CAPITALIZED] [ATTRIBUTE] XSD.ID,

respondWith-list [UNTAGGED] SEQUENCE OF respondWith [NAME AS CAPITALIZED]

RespondWith,

signature [NAME AS CAPITALIZED] Signature OPTIONAL}

RespondWith ::= [ELEMENT] XSD.QName

Request ::= [ELEMENT] RequestType

RequestType ::= SEQUENCE {

issueInstant [NAME AS CAPITALIZED] [ATTRIBUTE] XSD.DateTime,
majorVersion [NAME AS CAPITALIZED] [ATTRIBUTE] INTEGER,
minorVersion [NAME AS CAPITALIZED] [ATTRIBUTE] INTEGER,
requestID [NAME AS CAPITALIZED] [ATTRIBUTE] XSD.ID,
respondWith-list [UNTAGGED] SEQUENCE OF respondWith [NAME AS CAPITALIZED]
RespondWith,
signature [NAME AS CAPITALIZED] Signature OPTIONAL,
choice [UNTAGGED] CHOICE {
query [NAME AS CAPITALIZED] Query,
subjectQuery [NAME AS CAPITALIZED] SubjectQuery,
authenticationQuery [NAME AS CAPITALIZED] AuthenticationQuery,
attributeQuery [NAME AS CAPITALIZED] AttributeQuery,
authorizationDecisionQuery [NAME AS CAPITALIZED]
AuthorizationDecisionQuery,
assertionIDReference-list [UNTAGGED] SEQUENCE (SIZE(1..MAX)) OF
assertionIDReference [NAME AS CAPITALIZED] AssertionIDReference,
assertionArtifact-list [UNTAGGED] SEQUENCE (SIZE(1..MAX)) OF
assertionArtifact [NAME AS CAPITALIZED] AssertionArtifact}}

AssertionArtifact ::= [ELEMENT] XSD.String

Query ::= [ELEMENT] QueryAbstractType-derivations

QueryAbstractType ::= SEQUENCE { }

SubjectQuery ::= [ELEMENT] SubjectQueryAbstractType-derivations

SubjectQueryAbstractType ::= SEQUENCE {

subject [NAME AS CAPITALIZED] Subject}

AuthenticationQuery ::= [ELEMENT] AuthenticationQueryType

```
AuthenticationQueryType ::= SEQUENCE {  
    authenticationMethod [NAME AS CAPITALIZED] [ATTRIBUTE] XSD.AnyURI OPTIONAL,  
    subject [NAME AS CAPITALIZED] Subject}
```

```
AttributeQuery ::= [ELEMENT] AttributeQueryType
```

```
AttributeQueryType ::= SEQUENCE {  
    resource [NAME AS CAPITALIZED] [ATTRIBUTE]  
        XSD.AnyURI OPTIONAL,  
    subject [NAME AS CAPITALIZED] Subject,  
    attributeDesignator-list [UNTAGGED] SEQUENCE OF attributeDesignator  
        [NAME AS CAPITALIZED] AttributeDesignator}
```

```
AuthorizationDecisionQuery ::= [ELEMENT] AuthorizationDecisionQueryType
```

```
AuthorizationDecisionQueryType ::= SEQUENCE {  
    resource [NAME AS CAPITALIZED] [ATTRIBUTE] XSD.AnyURI,  
    subject [NAME AS CAPITALIZED] Subject,  
    action-list [UNTAGGED] SEQUENCE (SIZE(1..MAX)) OF action  
        [NAME AS CAPITALIZED] Action,  
    evidence [NAME AS CAPITALIZED] Evidence OPTIONAL}
```

```
ResponseAbstractType ::= SEQUENCE {  
    inResponseTo [NAME AS CAPITALIZED] [ATTRIBUTE] XSD.NCName OPTIONAL,  
    issueInstant [NAME AS CAPITALIZED] [ATTRIBUTE] XSD.DateTime,  
    majorVersion [NAME AS CAPITALIZED] [ATTRIBUTE] INTEGER,  
    minorVersion [NAME AS CAPITALIZED] [ATTRIBUTE] INTEGER,  
    recipient [NAME AS CAPITALIZED] [ATTRIBUTE] XSD.AnyURI OPTIONAL,  
    responseID [NAME AS CAPITALIZED] [ATTRIBUTE] XSD.ID,  
    signature [NAME AS CAPITALIZED] Signature OPTIONAL}
```

```
Response ::= [ELEMENT] ResponseType
```

```
ResponseType ::= SEQUENCE {  
    inResponseTo [NAME AS CAPITALIZED] [ATTRIBUTE] XSD.NCName OPTIONAL,  
    issueInstant [NAME AS CAPITALIZED] [ATTRIBUTE] XSD.DateTime,  
    majorVersion [NAME AS CAPITALIZED] [ATTRIBUTE] INTEGER,
```

```
minorVersion    [NAME AS CAPITALIZED] [ATTRIBUTE] INTEGER,  
recipient       [NAME AS CAPITALIZED] [ATTRIBUTE] XSD.AnyURI OPTIONAL,  
responseID     [NAME AS CAPITALIZED] [ATTRIBUTE] XSD.ID,  
signature      [NAME AS CAPITALIZED] Signature OPTIONAL,  
status         [NAME AS CAPITALIZED] Status,  
assertion-list [UNTAGGED] SEQUENCE OF assertion  
                [NAME AS CAPITALIZED] Assertion}
```

Status ::= [ELEMENT] StatusType

```
StatusType ::= SEQUENCE {  
    statusCode    [NAME AS CAPITALIZED] StatusCode,  
    statusMessage [NAME AS CAPITALIZED] StatusMessage OPTIONAL,  
    statusDetail  [NAME AS CAPITALIZED] StatusDetail OPTIONAL}
```

StatusCode ::= [ELEMENT] StatusCodeType

```
StatusCodeType ::= SEQUENCE {  
    value        [NAME AS CAPITALIZED] [ATTRIBUTE] XSD.QName,  
    statusCode   [NAME AS CAPITALIZED] StatusCode OPTIONAL}
```

StatusMessage ::= [ELEMENT] XSD.String

StatusDetail ::= [ELEMENT] StatusDetailType

```
StatusDetailType ::= SEQUENCE {  
    elem-list [UNTAGGED] SEQUENCE OF elem [ANY-ELEMENT] UTF8String  
        (CONSTRAINED BY  
        { /* Shall conform to the "AnyElementFormat" specified in ITU-T  
          Rec. X.693 | ISO/IEC 8825-4 */ })}
```

```
QueryAbstractType-derivations ::= [USE-TYPE] CHOICE {  
    queryAbstractType    [NAME AS CAPITALIZED] QueryAbstractType,  
    attributeQueryType   [NAME AS CAPITALIZED] AttributeQueryType,  
    authenticationQueryType [NAME AS CAPITALIZED]  
        AuthenticationQueryType,  
    authorizationDecisionQueryType [NAME AS CAPITALIZED]  
        AuthorizationDecisionQueryType,
```


subjectQueryAbstractType [NAME AS CAPITALIZED]
SubjectQueryAbstractType}

SubjectQueryAbstractType-derivations ::= [USE-TYPE] CHOICE {
subjectQueryAbstractType [NAME AS CAPITALIZED]
SubjectQueryAbstractType,
attributeQueryType [NAME AS CAPITALIZED] AttributeQueryType,
authenticationQueryType [NAME AS CAPITALIZED]
AuthenticationQueryType,
authorizationDecisionQueryType [NAME AS CAPITALIZED]
AuthorizationDecisionQueryType}

ENCODING-CONTROL XER

GLOBAL-DEFAULTS MODIFIED-ENCODINGS

GLOBAL-DEFAULTS CONTROL-NAMESPACE

"http://www.w3.org/2001/XMLSchema-instance" PREFIX "xsi"

NAMESPACE ALL, RequestAbstractType.respondWith-list.respondWith,
RequestType.respondWith-list.respondWith, RequestType.choice.query,
RequestType.choice.subjectQuery,
RequestType.choice.authenticationQuery,
RequestType.choice.attributeQuery,
RequestType.choice.authorizationDecisionQuery,
RequestType.choice.assertionArtifact-list.assertionArtifact,
ResponseType.status, StatusType.statusCode, StatusType.statusMessage,
StatusType.statusDetail, StatusCodeType.statusCode,
QueryAbstractType-derivations.queryAbstractType,
QueryAbstractType-derivations.attributeQueryType,
QueryAbstractType-derivations.authenticationQueryType,
QueryAbstractType-derivations.authorizationDecisionQueryType,
QueryAbstractType-derivations.subjectQueryAbstractType,
SubjectQueryAbstractType-derivations.subjectQueryAbstractType,
SubjectQueryAbstractType-derivations.attributeQueryType,
SubjectQueryAbstractType-derivations.authenticationQueryType,
SubjectQueryAbstractType-derivations.authorizationDecisionQueryType
AS "urn:oasis:names:tc:SAML:1.0:protocol" PREFIX "samlp"
NAMESPACE RequestAbstractType.signature, RequestType.signature,
ResponseAbstractType.signature, ResponseType.signature
AS "http://www.w3.org/2000/09/xmldsig#" PREFIX "ds"

NAMESPACE

RequestType.choice.assertionIDReference-list.assertionIDReference,
SubjectQueryAbstractType.subject, AuthenticationQueryType.subject,
AttributeQueryType.subject,
AttributeQueryType.attributeDesignator-list.attributeDesignator,
AuthorizationDecisionQueryType.subject,
AuthorizationDecisionQueryType.action-list.action,
AuthorizationDecisionQueryType.evidence,
ResponseType.assertion-list.assertion
AS "urn:oasis:names:tc:SAML:1.0:assertion" PREFIX "saml"

END

XACML ASN.1 modules

/ cs-xacml-schema-content-01 */*

Context

DEFINITIONS XER INSTRUCTIONS AUTOMATIC TAGS ::=

BEGIN

IMPORTS

AnyURI, DateTime, String

FROM XSD */* Module standardized in [ITU-T Rec. X.694 | ISO/IEC 8825-5](#) */*

Obligations

FROM Policy; */* see below */*

Request ::= [ELEMENT] RequestType

RequestType ::= SEQUENCE {

subject-list [UNTAGGED] SEQUENCE (SIZE(1..MAX)) OF subject

[NAME AS CAPITALIZED] Subject,

resource [NAME AS CAPITALIZED] Resource,

action [NAME AS CAPITALIZED] Action,

environment [NAME AS CAPITALIZED] Environment OPTIONAL}

Response ::= [ELEMENT] ResponseType

ResponseType ::= SEQUENCE {

result-list [UNTAGGED] SEQUENCE (SIZE(1..MAX)) OF result

[NAME AS CAPITALIZED] Result}

Subject ::= [ELEMENT] SubjectType

SubjectType ::= SEQUENCE {
 subjectCategory [NAME AS CAPITALIZED] [NOT NAMESPACE] [ATTRIBUTE]
 XSD.AnyURI DEFAULT
 "urn:oasis:names:tc:xacml:1.0:subject-category:access-subject",
 attribute-list [UNTAGGED] SEQUENCE OF attribute
 [NAME AS CAPITALIZED] Attribute}

Resource ::= [ELEMENT] ResourceType

ResourceType ::= SEQUENCE {
 resourceContent [NAME AS CAPITALIZED] ResourceContent OPTIONAL,
 attribute-list [UNTAGGED] SEQUENCE OF attribute
 [NAME AS CAPITALIZED] Attribute}

ResourceContent ::= [ELEMENT] ResourceContentType

ResourceContentType ::= [EMBED-VALUES] SEQUENCE {
 embed-values SEQUENCE OF UTF8String,
 attr [ANY-ATTRIBUTES] SEQUENCE (CONSTRAINED BY
 {*/* Shall conform to the "AnyAttributeFormat" specified
 in ITU-T Rec. X.693 | ISO/IEC 8825-4 */*}) OF UTF8String,
 elem-list [UNTAGGED] SEQUENCE OF elem [ANY-ELEMENT] UTF8String
 (CONSTRAINED BY {*/* Shall conform to the "AnyElementFormat" specified
 in ITU-T Rec. X.693 | ISO/IEC 8825-4 */*})}
(CONSTRAINED BY {*/* Shall conform to ITU-T Rec. X.693 |
 ISO/IEC 8825-4, 25.2.5 */*})

Action ::= [ELEMENT] ActionType

ActionType ::= SEQUENCE {
 attribute-list [UNTAGGED] SEQUENCE OF attribute
 [NAME AS CAPITALIZED] Attribute}

Environment ::= [ELEMENT] EnvironmentType

```
EnvironmentType ::= SEQUENCE {  
    attribute-list [UNTAGGED] SEQUENCE OF attribute  
        [NAME AS CAPITALIZED] Attribute}
```

```
Attribute ::= [ELEMENT] AttributeType
```

```
AttributeType ::= SEQUENCE {  
    attributeId [NAME AS CAPITALIZED] [NOT NAMESPACE] [ATTRIBUTE] XSD.AnyURI,  
    dataType [NAME AS CAPITALIZED] [NOT NAMESPACE] [ATTRIBUTE] XSD.AnyURI,  
    issueInstant [NAME AS CAPITALIZED] [NOT NAMESPACE] [ATTRIBUTE]  
        XSD.DateTime OPTIONAL,  
    issuer [NAME AS CAPITALIZED] [NOT NAMESPACE] [ATTRIBUTE]  
        XSD.String OPTIONAL,  
    attributeValue [NAME AS CAPITALIZED] AttributeValue}
```

```
AttributeValue ::= [ELEMENT] AttributeValueType
```

```
AttributeValueType ::= [EMBED-VALUES] SEQUENCE {  
    embed-values SEQUENCE OF UTF8String,  
    attr [ANY-ATTRIBUTES] SEQUENCE (CONSTRAINED BY  
        {/* Shall conform to the "AnyAttributeFormat" specified  
            in ITU-T Rec. X.693 | ISO/IEC 8825-4 */}) OF UTF8String,  
    elem-list [UNTAGGED] SEQUENCE OF elem [ANY-ELEMENT] UTF8String  
        (CONSTRAINED BY {/* Shall conform to the "AnyElementFormat" specified  
            in ITU-T Rec. X.693 | ISO/IEC 8825-4 */})}  
(CONSTRAINED BY {/* Shall conform to ITU-T Rec. X.693 |  
            ISO/IEC 8825-4, 25.2.5 */})
```

```
Result ::= [ELEMENT] ResultType
```

```
ResultType ::= SEQUENCE {  
    resourceId [NAME AS CAPITALIZED] [NOT NAMESPACE] [ATTRIBUTE]  
        XSD.String OPTIONAL,  
    decision [NAME AS CAPITALIZED] Decision,  
    status [NAME AS CAPITALIZED] Status,  
    obligations [NAME AS CAPITALIZED] Obligations OPTIONAL}
```

Decision ::= [ELEMENT] DecisionType

DecisionType ::= ENUMERATED {
 deny, indeterminate, notApplicable, permit}

Status ::= [ELEMENT] StatusType

StatusType ::= SEQUENCE {
 statusCode [NAME AS CAPITALIZED] StatusCode,
 statusMessage [NAME AS CAPITALIZED] StatusMessage OPTIONAL,
 statusDetail [NAME AS CAPITALIZED] StatusDetail OPTIONAL}

StatusCode ::= [ELEMENT] StatusCodeType

StatusCodeType ::= SEQUENCE {
 value [NAME AS CAPITALIZED] [NOT NAMESPACE] [ATTRIBUTE] XSD.AnyURI,
 statusCode [NAME AS CAPITALIZED] StatusCode OPTIONAL}

StatusMessage ::= [ELEMENT] XSD.String

StatusDetail ::= [ELEMENT] StatusDetailType

StatusDetailType ::= SEQUENCE {
 elem-list [UNTAGGED] SEQUENCE OF elem [ANY-ELEMENT] UTF8String
 (CONSTRAINED BY { /* Shall conform to the "AnyElementFormat"
 specified in ITU-T Rec. X.693 | ISO/IEC 8825-4 */ }) }

ENCODING-CONTROL XER

GLOBAL-DEFAULTS MODIFIED-ENCODINGS

GLOBAL-DEFAULTS CONTROL-NAMESPACE

 "http://www.w3.org/2001/XMLSchema-instance" PREFIX "xsi"

NAMESPACE ALL, ALL IN ALL AS "urn:oasis:names:tc:xacml:1.0:context"

 PREFIX "xacml-context"

NAMESPACE ResultType.obligations

 AS "urn:oasis:names:tc:xacml:1.0:policy" PREFIX "xacml"

TEXT DecisionType:ALL AS CAPITALIZED

END

```
/* cs-xacml-schema-policy-01 */
Policy
DEFINITIONS XER INSTRUCTIONS AUTOMATIC TAGS ::=
BEGIN
IMPORTS
    AnyURI, String, AnyType
    FROM XSD;

PolicySet ::= [ELEMENT] PolicySetType

PolicySetType ::= SEQUENCE {
    policyCombiningAlgId [NAME AS CAPITALIZED] [NOT NAMESPACE] [ATTRIBUTE]
        XSD.AnyURI,
    policySetId           [NAME AS CAPITALIZED] [NOT NAMESPACE] [ATTRIBUTE]
        XSD.AnyURI,
    description           [NAME AS CAPITALIZED] Description OPTIONAL,
    policySetDefaults    [NAME AS CAPITALIZED] PolicySetDefaults OPTIONAL,
    target                [NAME AS CAPITALIZED] Target,
    choice-list           [UNTAGGED] SEQUENCE OF [UNTAGGED] CHOICE {
        policySet         [NAME AS CAPITALIZED] PolicySet,
        policy             [NAME AS CAPITALIZED] Policy,
        policySetIdReference [NAME AS CAPITALIZED] PolicySetIdReference,
        policyIdReference  [NAME AS CAPITALIZED] PolicyIdReference},
    obligations           [NAME AS CAPITALIZED] Obligations OPTIONAL}

PolicySetIdReference ::= [ELEMENT] XSD.AnyURI

PolicyIdReference ::= [ELEMENT] XSD.AnyURI

PolicySetDefaults ::= [ELEMENT] DefaultsType

PolicyDefaults ::= [ELEMENT] DefaultsType

DefaultsType ::= SEQUENCE {
    choice [UNTAGGED] CHOICE {
        XPathVersion [NAME AS CAPITALIZED] XPathVersion}
```

}

/* */

XPathVersion ::= [ELEMENT] XSD.AnyURI

/* */

Policy ::= [ELEMENT] PolicyType

PolicyType ::= SEQUENCE {
 policyId [NAME AS CAPITALIZED] [NOT NAMESPACE] [ATTRIBUTE]
 XSD.AnyURI,
 ruleCombiningAlgId [NAME AS CAPITALIZED] [NOT NAMESPACE] [ATTRIBUTE]
 XSD.AnyURI,
 description [NAME AS CAPITALIZED] Description OPTIONAL,
 policyDefaults [NAME AS CAPITALIZED] PolicyDefaults OPTIONAL,
 target [NAME AS CAPITALIZED] Target,
 rule-list [UNTAGGED] SEQUENCE OF rule [NAME AS CAPITALIZED] Rule,
 obligations [NAME AS CAPITALIZED] Obligations OPTIONAL}

Description ::= [ELEMENT] XSD.String

Rule ::= [ELEMENT] RuleType

RuleType ::= SEQUENCE {
 effect [NAME AS CAPITALIZED] [NOT NAMESPACE] [ATTRIBUTE] EffectType,
 ruleId [NAME AS CAPITALIZED] [NOT NAMESPACE] [ATTRIBUTE] XSD.AnyURI,
 description [NAME AS CAPITALIZED] Description OPTIONAL,
 target [NAME AS CAPITALIZED] Target OPTIONAL,
 condition [NAME AS CAPITALIZED] Condition OPTIONAL}

EffectType ::= ENUMERATED {
 deny, permit}

Target ::= [ELEMENT] TargetType

TargetType ::= SEQUENCE {

```
subjects [NAME AS CAPITALIZED] Subjects,  
resources [NAME AS CAPITALIZED] Resources,  
actions [NAME AS CAPITALIZED] Actions}
```

```
Subjects ::= [ELEMENT] SubjectsType
```

```
SubjectsType ::= SEQUENCE {  
  choice [UNTAGGED] CHOICE {  
    subject-list [UNTAGGED] SEQUENCE (SIZE(1..MAX)) OF subject  
      [NAME AS CAPITALIZED] Subject,  
    anySubject [NAME AS CAPITALIZED] AnySubject}}
```

```
Subject ::= [ELEMENT] SubjectType
```

```
SubjectType ::= SEQUENCE {  
  subjectMatch-list [UNTAGGED] SEQUENCE (SIZE(1..MAX)) OF subjectMatch  
    [NAME AS CAPITALIZED] SubjectMatch}
```

```
AnySubject ::= [ELEMENT] XSD.AnyType
```

```
Resources ::= [ELEMENT] ResourcesType
```

```
ResourcesType ::= SEQUENCE {  
  choice [UNTAGGED] CHOICE {  
    resource-list [UNTAGGED] SEQUENCE (SIZE(1..MAX)) OF resource  
      [NAME AS CAPITALIZED] Resource,  
    anyResource [NAME AS CAPITALIZED] AnyResource}}
```

```
AnyResource ::= [ELEMENT] XSD.AnyType
```

```
Resource ::= [ELEMENT] ResourceType
```

```
ResourceType ::= SEQUENCE {  
  resourceMatch-list [UNTAGGED] SEQUENCE (SIZE(1..MAX)) OF resourceMatch  
    [NAME AS CAPITALIZED] ResourceMatch}
```

```
Actions ::= [ELEMENT] ActionsType
```



```
ActionTypes ::= SEQUENCE {  
    choice [UNTAGGED] CHOICE {  
        action-list [UNTAGGED] SEQUENCE (SIZE(1..MAX)) OF action [NAME AS  
            CAPITALIZED] Action,  
        anyAction [NAME AS CAPITALIZED] AnyAction}}
```

```
AnyAction ::= [ELEMENT] XSD.AnyType
```

```
Action ::= [ELEMENT] ActionType
```

```
ActionType ::= SEQUENCE {  
    actionMatch-list [UNTAGGED] SEQUENCE (SIZE(1..MAX)) OF actionMatch  
        [NAME AS CAPITALIZED] ActionMatch}
```

```
SubjectMatch ::= [ELEMENT] SubjectMatchType
```

```
SubjectMatchType ::= SEQUENCE {  
    matchId [NAME AS CAPITALIZED] [NOT NAMESPACE] [ATTRIBUTE]  
        XSD.AnyURI,  
    attributeValue [NAME AS CAPITALIZED] AttributeValue,  
    choice [UNTAGGED] CHOICE {  
        subjectAttributeDesignator  
            [NAME AS CAPITALIZED] SubjectAttributeDesignator,  
        attributeSelector [NAME AS CAPITALIZED] AttributeSelector}}
```

```
ResourceMatch ::= [ELEMENT] ResourceMatchType
```

```
ResourceMatchType ::= SEQUENCE {  
    matchId [NAME AS CAPITALIZED] [NOT NAMESPACE] [ATTRIBUTE] XSD.AnyURI,  
    attributeValue [NAME AS CAPITALIZED] AttributeValue,  
    choice [UNTAGGED] CHOICE {  
        resourceAttributeDesignator  
            [NAME AS CAPITALIZED] ResourceAttributeDesignator,  
        attributeSelector [NAME AS CAPITALIZED] AttributeSelector}}
```

```
ActionMatch ::= [ELEMENT] ActionMatchType
```

```
ActionMatchType ::= SEQUENCE {
```

```
matchId          [NAME AS CAPITALIZED] [NOT NAMESPACE] [ATTRIBUTE] XSD.AnyURI,  
attributeValue  [NAME AS CAPITALIZED] AttributeValue,  
choice          [UNTAGGED] CHOICE {  
    actionAttributeDesignator [NAME AS CAPITALIZED]  
        ActionAttributeDesignator,  
    attributeSelector          [NAME AS CAPITALIZED] AttributeSelector}}
```

AttributeSelector ::= [ELEMENT] AttributeSelectorType

```
AttributeSelectorType ::= SEQUENCE {  
    dataType          [NAME AS CAPITALIZED] [NOT NAMESPACE] [ATTRIBUTE]  
        XSD.AnyURI,  
    mustBePresent    [NAME AS CAPITALIZED] [NOT NAMESPACE] [ATTRIBUTE]  
        BOOLEAN DEFAULT FALSE,  
    requestContextPath [NAME AS CAPITALIZED] [NOT NAMESPACE] [ATTRIBUTE]  
        XSD.String}
```

ResourceAttributeDesignator ::= [ELEMENT] AttributeDesignatorType-derivations

ActionAttributeDesignator ::= [ELEMENT] AttributeDesignatorType-derivations

EnvironmentAttributeDesignator ::= [ELEMENT]
AttributeDesignatorType-derivations

```
AttributeDesignatorType ::= SEQUENCE {  
    attributeId      [NAME AS CAPITALIZED] [NOT NAMESPACE] [ATTRIBUTE] XSD.AnyURI,  
    dataType         [NAME AS CAPITALIZED] [NOT NAMESPACE] [ATTRIBUTE] XSD.AnyURI,  
    issuer           [NAME AS CAPITALIZED] [NOT NAMESPACE] [ATTRIBUTE]  
        XSD.String OPTIONAL,  
    mustBePresent   [NAME AS CAPITALIZED] [NOT NAMESPACE] [ATTRIBUTE]  
        BOOLEAN DEFAULT FALSE}
```

SubjectAttributeDesignator ::= [ELEMENT] SubjectAttributeDesignatorType

```
SubjectAttributeDesignatorType ::= SEQUENCE {  
    attributeId      [NAME AS CAPITALIZED] [NOT NAMESPACE] [ATTRIBUTE]  
        XSD.AnyURI,  
    dataType         [NAME AS CAPITALIZED] [NOT NAMESPACE] [ATTRIBUTE]
```

```

XSD.AnyURI,
issuer          [NAME AS CAPITALIZED] [NOT NAMESPACE] [ATTRIBUTE]
                XSD.String OPTIONAL,
mustBePresent   [NAME AS CAPITALIZED] [NOT NAMESPACE] [ATTRIBUTE]
                BOOLEAN DEFAULT FALSE,
subjectCategory [NAME AS CAPITALIZED] [NOT NAMESPACE] [ATTRIBUTE]
                XSD.AnyURI DEFAULT
                "urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"}

```

AttributeValue ::= [ELEMENT] AttributeValueType-derivations

```

AttributeValueType ::= [EMBED-VALUES] SEQUENCE {
    embed-values SEQUENCE OF UTF8String,
    dataType     [NAME AS CAPITALIZED] [NOT NAMESPACE] [ATTRIBUTE] XSD.AnyURI,
    attr         [ANY-ATTRIBUTES] SEQUENCE (CONSTRAINED BY
        { /* Shall conform to the "AnyAttributeFormat" specified
            in ITU-T Rec. X.693 | ISO/IEC 8825-4 */ }) OF UTF8String,
    elem-list    [UNTAGGED] SEQUENCE OF elem [ANY-ELEMENT] UTF8String
        (CONSTRAINED BY { /* Shall conform to the "AnyElementFormat" specified
            in ITU-T Rec. X.693 | ISO/IEC 8825-4 */ }) }
(CONSTRAINED BY { /* Shall conform to ITU-T Rec. X.693 |
    ISO/IEC 8825-4, 25.2.5 */ })

```

Function ::= [ELEMENT] FunctionType

```

FunctionType ::= SEQUENCE {
    functionId [NAME AS CAPITALIZED] [NOT NAMESPACE] [ATTRIBUTE] XSD.AnyURI }

```

Apply ::= [ELEMENT] ApplyType

Condition ::= [ELEMENT] ApplyType

```

ApplyType ::= SEQUENCE {
    functionId [NAME AS CAPITALIZED] [NOT NAMESPACE] [ATTRIBUTE] XSD.AnyURI,
    choice-list [UNTAGGED] SEQUENCE OF [UNTAGGED] CHOICE {
        apply          [NAME AS CAPITALIZED] Apply,
        function       [NAME AS CAPITALIZED] Function,
        attributeValue [NAME AS CAPITALIZED] AttributeValue,

```

```
subjectAttributeDesignator    [NAME AS CAPITALIZED]
    SubjectAttributeDesignator,
resourceAttributeDesignator    [NAME AS CAPITALIZED]
    ResourceAttributeDesignator,
actionAttributeDesignator      [NAME AS CAPITALIZED]
    ActionAttributeDesignator,
environmentAttributeDesignator [NAME AS CAPITALIZED]
    EnvironmentAttributeDesignator,
attributeSelector              [NAME AS CAPITALIZED] AttributeSelector}}
```

Obligations ::= [ELEMENT] ObligationsType

```
ObligationsType ::= SEQUENCE {
    obligation-list [UNTAGGED] SEQUENCE (SIZE(1..MAX)) OF obligation
        [NAME AS CAPITALIZED] Obligation}
```

Obligation ::= [ELEMENT] ObligationType

```
ObligationType ::= SEQUENCE {
    fulfillOn                [NAME AS CAPITALIZED] [NOT NAMESPACE]
EffectType,
    obligationId              [NAME AS CAPITALIZED] [NOT NAMESPACE] [ATTRIBUTE]
        XSD.AnyURI,
    attributeAssignment-list [UNTAGGED] SEQUENCE (SIZE(1..MAX)) OF
        attributeAssignment [NAME AS CAPITALIZED] AttributeAssignment}
```

AttributeAssignment ::= [ELEMENT] AttributeAssignmentType

```
AttributeAssignmentType ::= [EMBED-VALUES] SEQUENCE {
    embed-values SEQUENCE OF UTF8String,
    attributeId [NAME AS CAPITALIZED] [NOT NAMESPACE] [ATTRIBUTE] XSD.AnyURI,
    dataType    [NAME AS CAPITALIZED] [NOT NAMESPACE] [ATTRIBUTE] XSD.AnyURI,
    attr        [ANY-ATTRIBUTES] SEQUENCE (CONSTRAINED BY
        {/* Shall conform to the "AnyAttributeFormat" specified
            in ITU-T Rec. X.693 | ISO/IEC 8825-4 */}) OF UTF8String,
    elem-list   [UNTAGGED] SEQUENCE OF elem [ANY-ELEMENT] UTF8String
        (CONSTRAINED BY {/* Shall conform to the "AnyElementFormat" specified
            in ITU-T Rec. X.693 | ISO/IEC 8825-4 */})}
(CONSTRAINED BY {/* Shall conform to ITU-T Rec. X.693 |
```

ISO/IEC 8825-4, 25.2.5 */})

```
AttributeDesignatorType-derivations ::= [USE-TYPE] CHOICE {
    attributeDesignatorType      [NAME AS CAPITALIZED]
        AttributeDesignatorType,
    subjectAttributeDesignatorType [NAME AS CAPITALIZED]
        SubjectAttributeDesignatorType}
```

```
AttributeValueType-derivations ::= [USE-TYPE] CHOICE {
    attributeValueType      [NAME AS CAPITALIZED] AttributeValueType,
    attributeAssignmentType [NAME AS CAPITALIZED] AttributeAssignmentType}
```

ENCODING-CONTROL XER

GLOBAL-DEFAULTS MODIFIED-ENCODINGS

GLOBAL-DEFAULTS CONTROL-NAMESPACE

"http://www.w3.org/2001/XMLSchema-instance" PREFIX "xsi"

NAMESPACE ALL, ALL IN ALL AS "urn:oasis:names:tc:xacml:1.0:policy"

PREFIX "xacml"

TEXT EffectType:ALL AS CAPITALIZED

END

Other ASN.1 modules

```
/* xmldsig-core-schema */
/* <!DOCTYPE schema
    PUBLIC "-//W3C//DTD XMLSchema 200102//EN"
    "http://www.w3.org/2001/XMLSchema.dtd"
    [
        <!ATTLIST schema
            xmlns:ds CDATA #FIXED "http://www.w3.org/2000/09/xmldsig#">
        <!ENTITY dsig 'http://www.w3.org/2000/09/xmldsig#'>
        <!ENTITY % p ''>
        <!ENTITY % s ''>
    ]> */
/* Schema for XML Signatures
    http://www.w3.org/2000/09/xmldsig#
```

\$Revision: 1.1 \$ on \$Date: 2002/02/08 20:32:26 \$ by \$Author: reagle \$

Copyright 2001 The Internet Society and W3C (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University). All Rights Reserved.

<http://www.w3.org/Consortium/Legal/>

This document is governed by the W3C Software License [1] as described in the FAQ [2].

[1] <http://www.w3.org/Consortium/Legal/copyright-software-19980720>

[2] <http://www.w3.org/Consortium/Legal/IPR-FAQ-20000620.html#DTD>

**/*

Xmldsig

DEFINITIONS XER INSTRUCTIONS AUTOMATIC TAGS ::=

BEGIN

IMPORTS

 ID, AnyURI, String

 FROM XSD; */* Module standardized in [ITU-T Rec. X.694 | ISO/IEC 8825-5](#) */*

/ Basic Types Defined for Signatures */*

CryptoBinary ::= [BASE64] OCTET STRING

/ Start Signature */*

Signature ::= [ELEMENT] SignatureType

SignatureType ::= SEQUENCE {

 id [NAME AS CAPITALIZED] [NOT NAMESPACE] [ATTRIBUTE]
 XSD.ID OPTIONAL,

 signedInfo [NAME AS CAPITALIZED] SignedInfo,

 signatureValue [NAME AS CAPITALIZED] SignatureValue,

 keyInfo [NAME AS CAPITALIZED] KeyInfo OPTIONAL,

 object-list [UNTAGGED] SEQUENCE OF object [NAME AS CAPITALIZED] Object}

SignatureValue ::= [ELEMENT] SignatureValueType

```
SignatureValueType ::= SEQUENCE {  
    id [NAME AS CAPITALIZED] [NOT NAMESPACE] [ATTRIBUTE] XSD.ID OPTIONAL,  
    base [BASE64] [UNTAGGED] OCTET STRING}
```

```
/* Start SignedInfo */
```

```
SignedInfo ::= [ELEMENT] SignedInfoType
```

```
SignedInfoType ::= SEQUENCE {  
    id [NAME AS CAPITALIZED] [NOT NAMESPACE] [ATTRIBUTE]  
        XSD.ID OPTIONAL,  
    canonicalizationMethod [NAME AS CAPITALIZED] CanonicalizationMethod,  
    signatureMethod [NAME AS CAPITALIZED] SignatureMethod,  
    reference-list [UNTAGGED] SEQUENCE (SIZE(1..MAX)) OF reference  
        [NAME AS CAPITALIZED] Reference}
```

```
CanonicalizationMethod ::= [ELEMENT] CanonicalizationMethodType
```

```
CanonicalizationMethodType ::= [EMBED-VALUES] SEQUENCE {  
    embed-values SEQUENCE OF UTF8String,  
    algorithm [NAME AS CAPITALIZED] [NOT NAMESPACE] [ATTRIBUTE] XSD.AnyURI,  
    elem-list [UNTAGGED] SEQUENCE OF elem [ANY-ELEMENT] UTF8String  
        (CONSTRAINED BY {/* Shall conform to the "AnyElementFormat" specified  
            in ITU-T Rec. X.693 | ISO/IEC 8825-4 */})}  
(CONSTRAINED BY {/* Shall conform to ITU-T Rec. X.693 |  
            ISO/IEC 8825-4, 25.2.5 */})
```

```
SignatureMethod ::= [ELEMENT] SignatureMethodType
```

```
SignatureMethodType ::= [EMBED-VALUES] SEQUENCE {  
    embed-values SEQUENCE OF UTF8String,  
    algorithm [NAME AS CAPITALIZED] [NOT NAMESPACE] [ATTRIBUTE]  
        XSD.AnyURI,  
    HMACOutputLength [NAME AS CAPITALIZED] HMACOutputLengthType OPTIONAL,  
    elem-list [UNTAGGED] SEQUENCE OF elem [ANY-ELEMENT EXCEPT ABSENT  
        "http://www.w3.org/2000/09/xmldsig#"] UTF8String  
        (CONSTRAINED BY {/* Shall conform to the "AnyElementFormat"  
            specified in ITU-T Rec. X.693 | ISO/IEC 8825-4 */})}
```

. (CONSTRAINED BY { /* Shall conform to ITU-T Rec. X.693 |
ISO/IEC 8825-4, 25.2.5 */ })

/* Start Reference */

Reference ::= [ELEMENT] ReferenceType

ReferenceType ::= SEQUENCE {
 id [NAME AS CAPITALIZED] [NOT NAMESPACE] [ATTRIBUTE]
 XSD.ID OPTIONAL,
 type [NAME AS CAPITALIZED] [NOT NAMESPACE] [ATTRIBUTE]
 XSD.AnyURI OPTIONAL,
 uRI [NAME AS CAPITALIZED] [NOT NAMESPACE] [ATTRIBUTE]
 XSD.AnyURI OPTIONAL,
 transforms [NAME AS CAPITALIZED] Transforms OPTIONAL,
 digestMethod [NAME AS CAPITALIZED] DigestMethod,
 digestValue [NAME AS CAPITALIZED] DigestValue}

Transforms ::= [ELEMENT] TransformsType

TransformsType ::= SEQUENCE {
 transform-list [UNTAGGED] SEQUENCE (SIZE(1..MAX)) OF transform
 [NAME AS CAPITALIZED] Transform}

Transform ::= [ELEMENT] TransformType

TransformType ::= [EMBED-VALUES] SEQUENCE {
 embed-values SEQUENCE OF UTF8String,
 algorithm [NAME AS CAPITALIZED] [NOT NAMESPACE] [ATTRIBUTE] XSD.AnyURI,
 choice-list [UNTAGGED] SEQUENCE OF [UNTAGGED] CHOICE {
 elem [ANY-ELEMENT EXCEPT ABSENT "http://www.w3.org/2000/09/xmlsig#"]
 UTF8String (CONSTRAINED BY { /* Shall conform to the
 "AnyElementFormat" specified in ITU-T Rec. X.693 |
 ISO/IEC 8825-4 */ }),
 XPath [NAME AS CAPITALIZED] XSD.String}}
(CONSTRAINED BY { /* Shall conform to ITU-T Rec. X.693 |
ISO/IEC 8825-4, 25.2.5 */ })

/ End Reference */*

DigestMethod ::= [ELEMENT] DigestMethodType

DigestMethodType ::= [EMBED-VALUES] SEQUENCE {
 embed-values SEQUENCE OF UTF8String,
 algorithm [NAME AS CAPITALIZED] [NOT NAMESPACE] [ATTRIBUTE] XSD.AnyURI,
 elem-list [UNTAGGED] SEQUENCE OF elem [ANY-ELEMENT EXCEPT ABSENT
 "http://www.w3.org/2000/09/xmldsig#"] UTF8String
 (CONSTRAINED BY {*/* Shall conform to the "AnyElementFormat" specified
 in ITU-T Rec. X.693 | ISO/IEC 8825-4 */*})}
(CONSTRAINED BY {*/* Shall conform to ITU-T Rec. X.693 |
 ISO/IEC 8825-4, 25.2.5 */*})

DigestValue ::= [ELEMENT] DigestValueType

DigestValueType ::= [BASE64] OCTET STRING

/ End SignedInfo */*

/ Start KeyInfo */*

KeyInfo ::= [ELEMENT] KeyInfoType

KeyInfoType ::= [EMBED-VALUES] SEQUENCE {
 embed-values SEQUENCE OF UTF8String,
 id [NAME AS CAPITALIZED] [NOT NAMESPACE] [ATTRIBUTE]
 XSD.ID OPTIONAL,
 choice-list [UNTAGGED] SEQUENCE (SIZE(1..MAX)) OF [UNTAGGED] CHOICE {
 keyName [NAME AS CAPITALIZED] KeyName,
 keyValue [NAME AS CAPITALIZED] KeyValue,
 retrievalMethod [NAME AS CAPITALIZED] RetrievalMethod,
 x509Data [NAME AS CAPITALIZED] X509Data,
 pgpData [NAME AS CAPITALIZED] PGPData,
 spkiData [NAME AS CAPITALIZED] SPKIData,
 mgmtData [NAME AS CAPITALIZED] MgmtData,
 elem [ANY-ELEMENT EXCEPT ABSENT
 "http://www.w3.org/2000/09/xmldsig#"] UTF8String

(CONSTRAINED BY {*/* Shall conform to the "AnyElementFormat" specified in ITU-T Rec. X.693 | ISO/IEC 8825-4 */*})}}

(CONSTRAINED BY {*/* Shall conform to ITU-T Rec. X.693 | ISO/IEC 8825-4, 25.2.5 */*})

KeyName ::= [ELEMENT] XSD.String

MgmtData ::= [ELEMENT] XSD.String

KeyValue ::= [ELEMENT] KeyValue

KeyValue ::= [EMBED-VALUES] SEQUENCE {
 embed-values SEQUENCE OF UTF8String,
 choice [UNTAGGED] CHOICE {
 dSAKeyValue [NAME AS CAPITALIZED] DSAKeyValue,
 rSAKeyValue [NAME AS CAPITALIZED] RSAKeyValue,
 elem [ANY-ELEMENT EXCEPT ABSENT
 "http://www.w3.org/2000/09/xmldsig#"] UTF8String
 (CONSTRAINED BY {*/* Shall conform to the "AnyElementFormat" specified in ITU-T Rec. X.693 | ISO/IEC 8825-4 */*})}}

(CONSTRAINED BY {*/* Shall conform to ITU-T Rec. X.693 | ISO/IEC 8825-4, 25.2.5 */*})

RetrievalMethod ::= [ELEMENT] RetrievalMethod

RetrievalMethod ::= SEQUENCE {
 type [NAME AS CAPITALIZED] [NOT NAMESPACE] [ATTRIBUTE]
 XSD.AnyURI OPTIONAL,
 uRI [NAME AS CAPITALIZED] [NOT NAMESPACE] [ATTRIBUTE]
 XSD.AnyURI OPTIONAL,
 transforms [NAME AS CAPITALIZED] Transforms OPTIONAL}

/ Start X509Data */*

X509Data ::= [ELEMENT] X509Data

X509Data ::= SEQUENCE {
 sequence-list [UNTAGGED] SEQUENCE (SIZE(1..MAX)) OF [UNTAGGED] SEQUENCE {

```
choice [UNTAGGED] CHOICE {
    x509IssuerSerial [NAME AS CAPITALIZED] X509IssuerSerialType,
    x509SKI           [NAME AS CAPITALIZED] [BASE64] OCTET STRING,
    x509SubjectName  [NAME AS CAPITALIZED] XSD.String,
    x509Certificate  [NAME AS CAPITALIZED] [BASE64] OCTET STRING,
    x509CRL          [NAME AS CAPITALIZED] [BASE64] OCTET STRING,
    elem             [ANY-ELEMENT EXCEPT ABSENT
        "http://www.w3.org/2000/09/xmldsig#"] UTF8String
        (CONSTRAINED BY { /* Shall conform to the "AnyElementFormat"
            specified in ITU-T Rec. X.693 | ISO/IEC 8825-4 */ }) } }
```

```
X509IssuerSerialType ::= SEQUENCE {
    x509IssuerName  [NAME AS CAPITALIZED] XSD.String,
    x509SerialNumber [NAME AS CAPITALIZED] INTEGER }
```

```
/* End X509Data */
```

```
/* Begin PGPData */
```

```
PGPData ::= [ELEMENT] PGPDataType
```

```
PGPDataType ::= SEQUENCE {
    choice [UNTAGGED] CHOICE {
        sequence [UNTAGGED] SEQUENCE {
            pGPKeyID [NAME AS CAPITALIZED] [BASE64] OCTET STRING,
            pGPKeyPacket [NAME AS CAPITALIZED] [BASE64] OCTET STRING OPTIONAL,
            elem-list [UNTAGGED] SEQUENCE OF elem
                [ANY-ELEMENT EXCEPT ABSENT
                    "http://www.w3.org/2000/09/xmldsig#"] UTF8String
                    (CONSTRAINED BY { /* Shall conform to the "AnyElementFormat"
                        specified in ITU-T Rec. X.693 | ISO/IEC 8825-4 */ }) },
            sequence-1 [UNTAGGED] SEQUENCE {
                pGPKeyPacket [NAME AS CAPITALIZED] [BASE64] OCTET STRING,
                elem-list [UNTAGGED] SEQUENCE OF elem [ANY-ELEMENT EXCEPT ABSENT
                    "http://www.w3.org/2000/09/xmldsig#"] UTF8String
                    (CONSTRAINED BY { /* Shall conform to the "AnyElementFormat"
                        specified in ITU-T Rec. X.693 | ISO/IEC 8825-4 */ }) } } }
```

/ End PGPDData */*

/ Begin SPKIData */*

SPKIData ::= [ELEMENT] SPKIDataType

SPKIDataType ::= SEQUENCE {

sequence-list [UNTAGGED] SEQUENCE (SIZE(1..MAX)) OF [UNTAGGED] SEQUENCE {

SPKISexp [NAME AS CAPITALIZED] [BASE64] OCTET STRING,

elem [ANY-ELEMENT EXCEPT ABSENT

"http://www.w3.org/2000/09/xmlldsig#"] UTF8String

(CONSTRAINED BY {*/* Shall conform to the "AnyElementFormat" specified
in ITU-T Rec. X.693 | ISO/IEC 8825-4 */*}) OPTIONAL}}

/ End SPKIData */*

/ End KeyInfo */*

/ Start Object (Manifest, SignatureProperty) */*

Object ::= [ELEMENT] ObjectType

ObjectType ::= [EMBED-VALUES] SEQUENCE {

embed-values SEQUENCE OF UTF8String,

encoding [NAME AS CAPITALIZED] [NOT NAMESPACE] [ATTRIBUTE]
XSD.AnyURI OPTIONAL,

id [NAME AS CAPITALIZED] [NOT NAMESPACE] [ATTRIBUTE]
XSD.ID OPTIONAL,

mimeType [NAME AS CAPITALIZED] [NOT NAMESPACE] [ATTRIBUTE]
XSD.String OPTIONAL,

sequence-list [UNTAGGED] SEQUENCE OF [UNTAGGED] SEQUENCE {

elem [ANY-ELEMENT] UTF8String (CONSTRAINED BY

{*/* Shall conform to the "AnyElementFormat" specified in
ITU-T Rec. X.693 | ISO/IEC 8825-4 */*})}}

(CONSTRAINED BY {*/* Shall conform to ITU-T Rec. X.693 |
ISO/IEC 8825-4, 25.2.5 */*})

Manifest ::= [ELEMENT] ManifestType

```
ManifestType ::= SEQUENCE {
    id [NAME AS CAPITALIZED] [NOT NAMESPACE] [ATTRIBUTE]
        XSD.ID OPTIONAL,
    reference-list [UNTAGGED] SEQUENCE (SIZE(1..MAX)) OF reference
        [NAME AS CAPITALIZED] Reference}

SignatureProperties ::= [ELEMENT] SignaturePropertiesType

SignaturePropertiesType ::= SEQUENCE {
    id [NAME AS CAPITALIZED] [NOT NAMESPACE] [ATTRIBUTE] XSD.ID OPTIONAL,
    signatureProperty-list [UNTAGGED] SEQUENCE (SIZE(1..MAX)) OF
        signatureProperty [NAME AS CAPITALIZED] SignatureProperty}

SignatureProperty ::= [ELEMENT] SignaturePropertyType

SignaturePropertyType ::= [EMBED-VALUES] SEQUENCE {
    embed-values SEQUENCE OF UTF8String,
    id [NAME AS CAPITALIZED] [NOT NAMESPACE] [ATTRIBUTE]
        XSD.ID OPTIONAL,
    target [NAME AS CAPITALIZED] [NOT NAMESPACE] [ATTRIBUTE] XSD.AnyURI,
    choice-list [UNTAGGED] SEQUENCE (SIZE(1..MAX)) OF [UNTAGGED] CHOICE {
        elem [ANY-ELEMENT EXCEPT ABSENT "http://www.w3.org/2000/09/xmldsig#"]
            UTF8String (CONSTRAINED BY {/* Shall conform to the
                "AnyElementFormat" specified in ITU-T Rec. X.693 |
                ISO/IEC 8825-4 */})})}
    (CONSTRAINED BY {/* Shall conform to ITU-T Rec. X.693 |
        ISO/IEC 8825-4, 25.2.5 */})

/* End Object (Manifest, SignatureProperty) */

/* Start Algorithm Parameters */

HMACOutputLengthType ::= INTEGER

/* Start KeyValue Element-types */

DSAKeyValue ::= [ELEMENT] DSAKeyValueType
```

```
DSAKeyValue ::= SEQUENCE {
    sequence [UNTAGGED] SEQUENCE {
        p [NAME AS CAPITALIZED] CryptoBinary,
        q [NAME AS CAPITALIZED] CryptoBinary} OPTIONAL,
    g [NAME AS CAPITALIZED] CryptoBinary OPTIONAL,
    y [NAME AS CAPITALIZED] CryptoBinary,
    j [NAME AS CAPITALIZED] CryptoBinary OPTIONAL,
    sequence-1 [UNTAGGED] SEQUENCE {
        seed [NAME AS CAPITALIZED] CryptoBinary,
        pgenCounter [NAME AS CAPITALIZED] CryptoBinary} OPTIONAL}

```

```
RSAKeyValue ::= [ELEMENT] RSAKeyValue
```

```
RSAKeyValue ::= SEQUENCE {
    modulus [NAME AS CAPITALIZED] CryptoBinary,
    exponent [NAME AS CAPITALIZED] CryptoBinary}

```

```
/* End KeyValue Element-types */
```

```
/* End Signature */
```

```
ENCODING-CONTROL XER
```

```
GLOBAL-DEFAULTS MODIFIED-ENCODINGS
```

```
GLOBAL-DEFAULTS CONTROL-NAMESPACE
```

```
"http://www.w3.org/2001/XMLSchema-instance" PREFIX "xsi"
```

```
NAMESPACE ALL, ALL IN ALL AS "http://www.w3.org/2000/09/xmldsig#"
```

```
PREFIX "ds"
```

```
END
```
